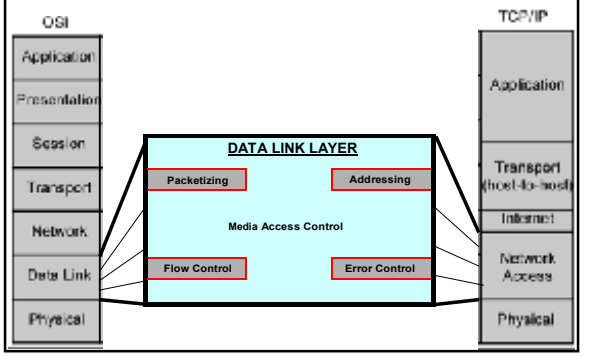
CPS 422
Computer Networks

DATA LINK LAYER

THE PERSPECTIVE

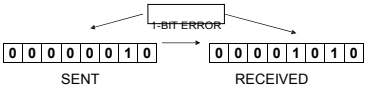


Data can be corrupted during transmission, for reliable communication errors must be detected and corrected

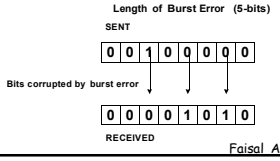
ERROR DETECTION AND CORRECTION

TYPES OF ERROR

‰ SINGLE - BIT ERRORS



‰ BURST ERRORS



ERROR DETECTION

‰ One error detection mechanism could be the transmission of all data units, TWICE.

‰ Error detection uses the concept of "Redundancy", which means adding extra bits to data for detecting errors at dest.

‰ The receiver puts the entire stream [data + redundancy] through a checking function

‰ If the bit stream passes checking criteria, the data unit is accepted and redundant bits discarded.

ERROR DETECTION (Contd…)

- Types of error detection techniques
  - Parity Check
    - Simple
    - Two-Dimensional
  - Cyclic Redundancy Check (CRC)
  - Checksum

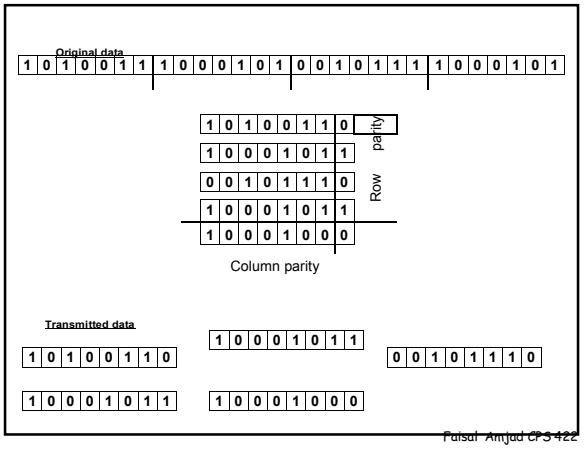
Simple Parity Check

- A redundant "Parity-bit" added to every data unit
- In "Even" parity, the total number of 1s including the parity bit must be even.
- In "Odd" parity, the total number of 1s including the parity bit must be Odd

Two-Dimensional Parity Check

- Block of bits is organized in a table (rows and columns)
  - Parity bits for each row and each column

  calculated and appended to the end of respective row/column.



Cyclic Redundancy Check (CRC)

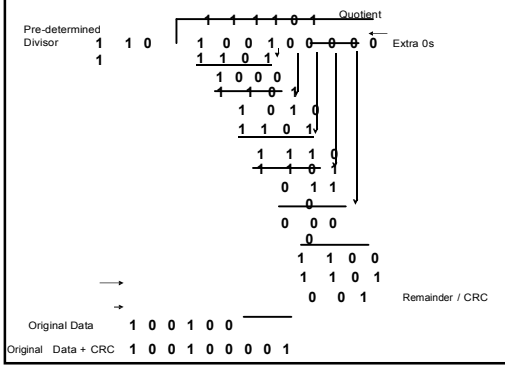
- Based on "binary Division"
- A sequence of redundant bits, called the CRC Remainder or simply "CRC" is appended to the end of data unit.
- Resulting data unit becomes exactly divisible by a second, "predetermined" binary number called the "Divisor".

CRC – Process

- A string of n 0s (zeros) appended to the data unit.
- The divisor is n+1 bits long.
- Elongated data unit is divided by the divisor, using binary division. Remainder of this process is the CRC.
- The CRC replaces the n 0s appended earlier at the beginning, and bit stream transmitted.
- The CRC may consist of all 0s.
- The receiver divides the received stream by the Divisor. If remainder is all 0s, no error occurred

http://www.uandiStar.org

CRC - Generation



CRC - Checking

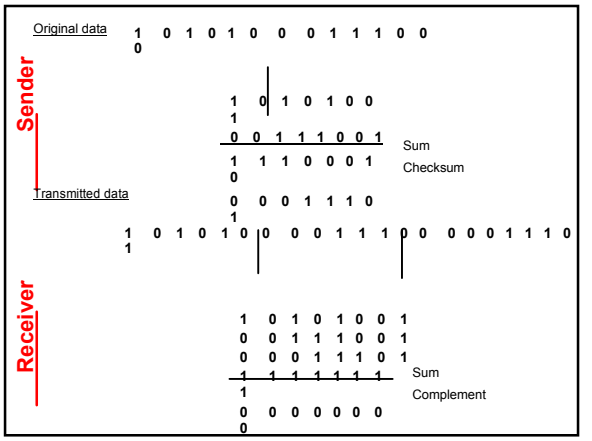
Similar process – Remainder must be all 0s, for data to be correct

CRC Divisor is represented as a polynomial e.g. in our example its equivalent polynomial is $x^3 + x^2 + 1$ (1101)

CHECKSUM

- Steps at Sender
  - The unit is divided into k sections, each n bits long
  - All sections are added using "1's Compliment"
  - The resulting sum is complemented which becomes the "Checksum"
  - Checksum is transmitted with the data.
- Steps at Receiver
  - The unit is divided into k sections, each n bits long
  - All sections are added using "1's Compliment"
  - The resulting sum is complemented
  - If the result is zero, data is accepted.

Checksum - Generation



ASSIGNMENT # 2

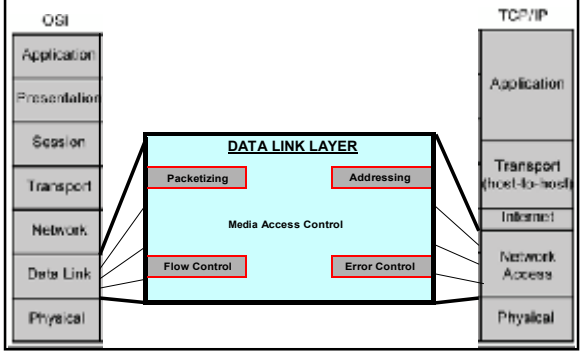
- Write a note on the performance of fol:
  - Simple Parity check
  - Two-Dimensional parity check
  - CRC
  - Checksum

http://www.uandiStar.org

CPS 422
Computer Networks

DATA LINK LAYER

THE PERSPECTIVE



ERROR CORRECTION

ERROR CORRECTION BY RETRANSMISSION

FORWARD ERROR CORRECTION

FORWARD ERROR CORRECTION

- A Receiver can use error correcting codes to automatically correct certain errors.
- Theoretically any number of errors can be corrected.
- To "Correct" an error the receiver must know the "location" of error in received data.

FORWARD ERROR CORRECTION (Contd…)

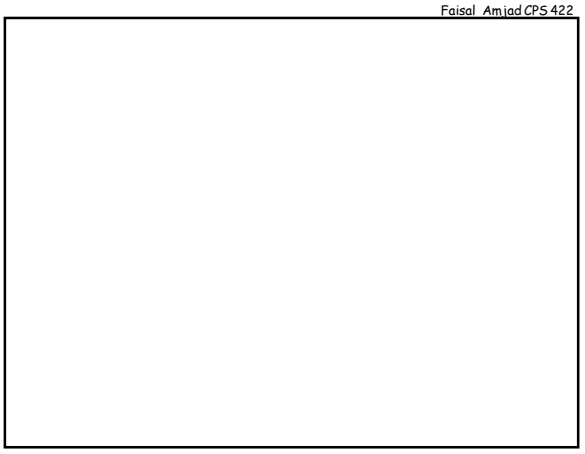
- To correct a "single-bit" error in an ASCII character, the error correcting code must determine which of the 7 bits has changed.
- So it has to distinguish between eight different states:
  - No error
  - Error in position 1
  - Error in position 2
  - …
  - Error in position 7
- It "seems" 3 redundant bits are required to represent these 8 states (000 – 111).

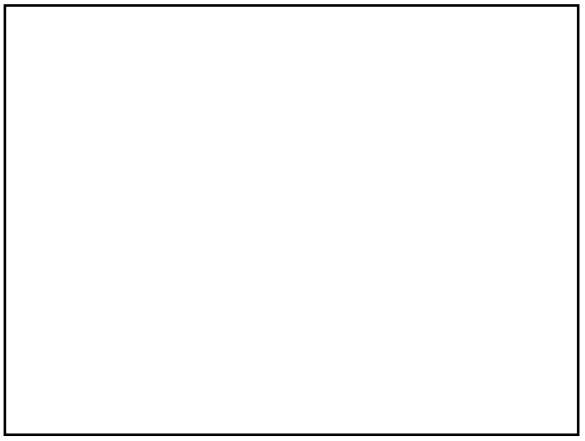
FORWARD ERROR CORRECTION (Contd…)

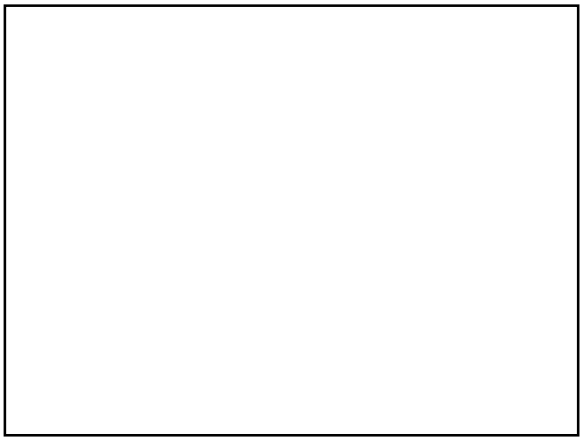
- What if an error occurs in redundancy bits???
- So we have to cater for 10 bits (7 data + 3 redundant)
- Hence 3 bits are insufficient for 7 bit data.
- To calculate "r" redundant bits for "m" data bits, consider this:
  - r must be able to indicate m + r +1 states. Why????
  - For m + r **error** states and 1 **no-error** state.
- Since r can represent 2^r states, so 2^r must be greater than or equal to m+r+1

- $2^r >= m + r + 1$

Faisal Amjad CPS 422



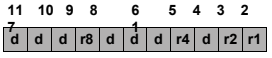




http://www.uandiStar.org

THE HAMMING CODE

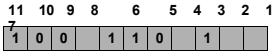
- Can be applied to data units of any length and uses the same relationship between data and redundant bits.
- Redundant bits are interspersed with data bits at positions which are powers of 2, e.g. 1,2,4,8...
- These bits are represented as r1,r2,r4,r8 etc.



THE HAMMING CODE

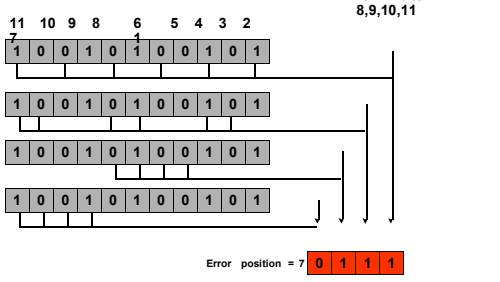
- Each redundant r bit is the parity bit for one combination of data bits as fol :-
  - r1 bits 1,3,5,7,9,11
  - r2 bits 2,3,6,7,10,11
  - r4 bits 4,5,6,7
  - r8 bits 8,9,10,11
- Plug in the data bits at their respective locations
- Calculate each r bit as parity of the bit-combination mentioned above

Data bits are : 1001101



Error detection using Hamming code

Bit-combination
r1 bits 1,3,5,7,9,11
r2 bits 2,3,6,7,10,11
r4 bits 4,5,6,7
r8 bits 8,9,10,11



ASSIGNMENT # 2

- Write a note on the performance of fol:
  - Simple Parity check
  - Two-Dimensional parity check
  - CRC
  - Checksum
  - Hamming Code