Week 3 –
Data Link Layer Contd.

COMP90007
Internet Technologies

Error Bounds: Hamming distance

A code turns data of *n* bits into codewords of *n+k* bits

Hamming distance is the minimum bit flips to turn one valid codeword into any other valid one:

- Example with 4 codewords of 10 bits (n=2, k=8):
  - 0000000000,
  - 0000011111,
  - 1111100000,
  - 1111111111

Hamming distance is 5 here!

Bounds for a code with a given distance D is then:

- D=2d+1 – can correct d errors (e.g., 2 errors above)
- D=d+1 – can detect d errors (e.g., 4 errors above)

Error Bounds

Q: Why can a code with distance *2d+1 can* **correct** up to *d* errors only?

- Errors are corrected by mapping a received invalid codeword to the nearest valid codeword, i.e., the one that can be reached with the fewest bit flips
- If there are more than d bit flips, then the received codeword may be closer to another valid codeword than the codeword that was sent

Good Case: Sending 0000000000 with 2 flips might give 1100000000 which is closest to 0000000000, correcting the error

Bad Case: But with 3 flips 1110000000 might be received, which is closest to 1111100000, which is still an error

A More Advanced Method: Hamming Codes

- $n=2^k-k-1$ (n: number of data, k: check bits)

- Put **<u>check bits in positions *p* that are power of 2</u>**, starting with position 1

Hamming Codes Example

- Example: Data: 0101 - > requires 3 check bits

$$4 = (2^3) - 3 - 1$$

Example Contd

- Data: 0101 → requires 3 check bits

Position	P1	P2	P3	P4	P5	P6	P7
Data	?	?	0	?	1	0	1

Put check bits in positions p that are power of 2, starting with position 1

Then: Calculate the **parity** for P1, P2, P4

P1 + P3 + P5 + P7 = ?+0+1+1= 0 (even OK)

P2 + P3 + P6 + P7 = ?+0+0+1= 1 (odd ADD)

P4 + P5 + P6 + P7 = ?+1+0+1= 0 (even OK)

7 is 111 so appears in all formula

Example Contd

Position	P1	P2	P3	P4	P5	P6	P7
Data	?	?	0	?	1	0	1

Calculating the parity bits for P1, P2, P4

P1 + P3 + P5 + P7 = ?+0+1+1= 0

P2 + P3 + P6 + P7 = ?+0+0+1= 1

P4 + P5 + P6 + P7 = ?+1+0+1= 0

Data sent: 0100101

Example 1: At the receiver: 0100100 (error)	Example 2: At the receiver: 0000101 (error)
P1 + P3 + P5 + P7 = 0+0+1+0= 1 ×	P1 + P3 + P5 + P7 = 0+0+1+1= 0
P2 + P3 + P6 + P7 = 1+0+0+0= 1 ×	P2 + P3 + P6 + P7 = 0+0+0+1= 1 ×
P4 + P5 + P6 + P7 = 0+1+0+0= 1 ×	P4 + P5 + P6 + P7 = 0+1+0+1= 0
Error bit = P1+P2+P4 = P7	Error bit = P2

Example Contd

- The example we saw is in the category of *error correction*
- And the example design could correct *only single bit errors*