Error Detection and Correction

COMP476

Networked Computer Systems

Goals

- Understand how errors might be detected
- Understand how errors might be corrected
- Be able to estimate the overhead required for error detection and correction.

Errors!!!

- The networking field spends a lot of effort correcting errors
- When you add two numbers, you rarely worry if the computer will add correctly
- When data is sent from one computer to another, it can be corrupted, lost, delayed or duplicated

Sources of Signal Errors

- **Interference** – Electromagnetic radiation from the environment
- **Distortion** – Capacitance and induction of wires will distort the signal
- **Attenuation** – As a signal travels through the media, it slowly loses energy

Bursts of Errors

- Transmission errors may change an individual bit or they may change many bits at once
- An error burst is when a electrical disturbance causes things to go wrong for a short period of time
- Several bits in a row might be changed by an error burst

Detection and Correction

- Most networking systems concentrate on detecting an error
  - If an error occurs, the data is retransmitted
- Some system attempt to identify the bits that are incorrect and then correct them

Parity

- The simplest way to detect memory errors is to add a parity bit.
- The parity bit is computed and stored in an extra bit in memory.
- The parity bit is the XOR of the data bits.
- The parity bit is set to make the sum of all one bits in the data and parity an even number.

Parity Error Detection

- When data is sent, the parity is computed and sent at the end of the data.
- When data is received, the parity is calculated.
  - If the received parity bit is different from the calculated value, an error has occurred
  - If the received parity bit is the same as the calculated bit, the data ***might*** be OK

Parity Bits

data	parity
0100110100	0
1111000010	1
1111010010	0
1011000110	1

Error Correction

- By using multiple parity bits, it is possible to detect which bit is wrong.
- Since bits only have two possible values, if you know a bit is wrong, the other value must be right.

Block Parity

	D	D	D	D	D	D	D	D	P
D	0	0	1	0	0	1	1	1	0
D	1	1	0	1	1	1	0	1	0
D	1	1	0	1	0	1	0	1	1
D	1	0	0	0	0	1	0	1	1
P	1	0	1	0	1	0	1	0	0

Block Parity

	D	D	D	D	D	D	D	D	P
D	0	0	1	0	0	1	1	1	0
D	1	1	1	1	1	1	0	1	1
D	1	1	0	1	0	1	0	1	1
D	1	0	0	0	0	1	0	1	1
P	1	0	0	0	1	0	1	0	1

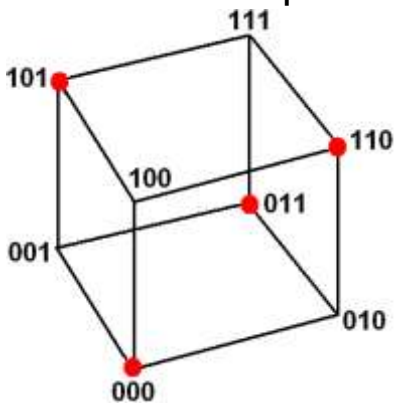
Hamming Distance

- Consider only one bit errors.
- In this table changing any one bit of a good code word converts it into an illegal code word.
- The Hamming distance between two words is the number of different bits.

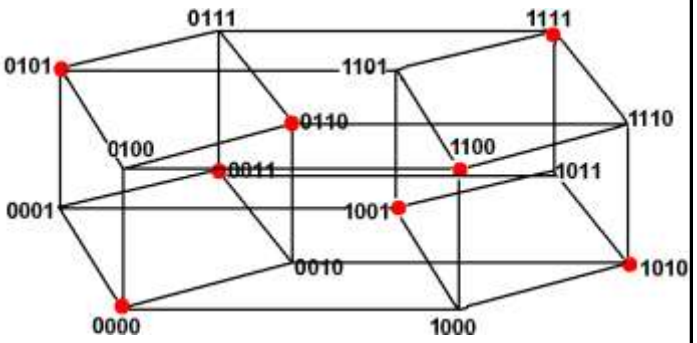
000	good
001	bad
011	good
010	bad
110	good
111	bad
101	good
100	bad

Gray code order

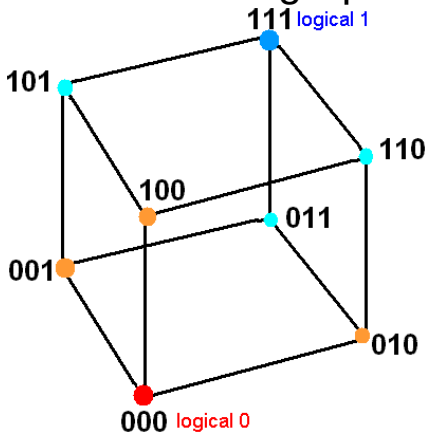
3 bit Data Space



4 bit Data Space



1 bit Correcting Space



Number of Correction Bits

$$d + p + 1 \leq 2^p$$

where:

- d is the number of data bits
- p is the number of check bits

approximately $\mathbf{p = \log_2(d) + 1}$

example:

for 32 data bits, you need 6 check bits

How many check bits are needed for an 8 bit byte?

1. 1
2. 2
3. 3
4. 4
5. 8

Hamming Codeword

P1	P2	D7	P4	D6	D5	D4	P8	D3	D2	D1	D0
1	1	1	0	0	1	0	1	1	0	1	1
1	2	3	4	5	6	7	8	9	10	11	12

P1 is XOR of 3, 5, 7, 9, 11

P2 is XOR of 3, 6, 7, 10, 11

P4 is XOR of 5, 6, 7, 12

P8 is XOR of 9, 10, 11, 12

Hamming Codeword

P1	P2	D7	P4	D6	D5	D4	P8	D3	D2	D1	D0
1	1	1	0	0	1	0	1	1	1	1	1
1	2	3	4	5	6	7	8	9	10	11	12

P1 is XOR of 3, 5, 7, 9, 11

P2 is XOR of 3, 6, 7, 10, 11

P4 is XOR of 5, 6, 7, 12

P8 is XOR of 9, 10, 11, 12

The table identifies the bad bit where 0 = correct parity and 1 = wrong parity

1	0	1	0
P8	P4	P2	P1

Hamming Codeword

P1	P2	D7	P4	D6	D5	D4	P8	D3	D2	D1	D0
1	1	0	0	0	1	0	1	1	0	1	1
1	2	3	4	5	6	7	8	9	10	11	12

P1 is XOR of 3, 5, 7, 9, 11

P2 is XOR of 3, 6, 7, 10, 11

P4 is XOR of 5, 6, 7, 12

P8 is XOR of 9, 10, 11, 12

The table identifies the bad bit where 0 = correct parity and 1 = wrong parity

0	0	1	1
P8	P4	P2	P1

Hamming Codeword

P1	P2	D7	P4	D6	D5	D4	P8	D3	D2	D1	D0
1	1	0	1	0	1	0	1	1	0	1	1
1	2	3	4	5	6	7	8	9	10	11	12

P1 is XOR of 3, 5, 7, 9, 11

P2 is XOR of 3, 6, 7, 10, 11

P4 is XOR of 5, 6, 7, 12

P8 is XOR of 9, 10, 11, 12

The table identifies the bad bit where 0 = correct parity and 1 = wrong parity

0	1	0	0
P8	P4	P2	P1

Checksum

- A checksum is another way to detect errors instead of using a parity bit
- When a packet of bytes is sent, the arithmetic sum of the bytes is sent at the end
- When a packet is received, the bytes are summed and compared to the value received.
- If the values are different, an error has occurred

Accuracy

- A parity bit is only one bit. It has two values.
- If multiple errors have occurred, the parity bits has a 50% chance of detecting the error
- A 16 bit checksum has one correct value and 65,535 wrong values.
- With multiple errors, a 16 bit checksum will be wrong only one in 64K times

If a byte of all zero bits is accidently sent, a checksum will

1. detect the error
2. miss the error

Cyclic Redundancy Checks

- A cyclic redundancy check is an improvement to a checksum
- It is based on polynomial division
- An n-bit CRC will detect any single error burst not longer than n bits
- CRCs are commonly 9, 17, 33 or 65 bits

Hardware CRC Calculation

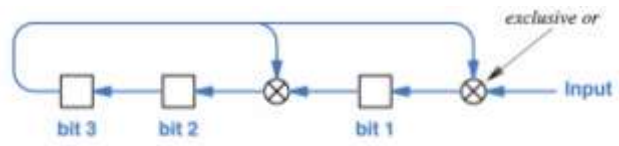


Figure 8.13 A hardware unit to compute a 3-bit CRC for $x^3 + x^1 + 1$.