Exercise Sensor Networks - (till March 31, 2008)

Lecture 3: Error recovery and energy efficient MAC

Exercise 3.1: CRC polynomials

Divide the message 10111010011 by the generator polynomial 10011 as done in the lecture. Write down the whole message as if it was transmitted to a receiver.

Solution:

```
101110100110000
10011
0010001
  10011
  00010001
     10011
     00010100
        10011
        0011100
          10011
          01111=rest
```

Verify:

```
101110100111111
10011
0010001
 10011
 00010001
   10011
    10111
    10011
     10011
     10011
       0=rest
```

XOR the rest with the extended message:

```
     101110100110000 message
 XOR             1111 rest
     101110100111111 result to transmit
```

Exercise Sensor Networks

Lecture 3: Error recovery and energy efficient MAC

Exercise 3.2: CRC polynomials

Write a function in Java or C which does the division above. The messages and the generator polynomials should be the input of the function (you can use strings of the kind “01001” but real bit operations are even more appreciated). The boolean result should denote if the message was divisible without a rest or not.

Solution:

```
bool Divisible(char* bit_string, long length_bit_string, char* generator, long length_generator)
{
  // generator is longer than bit_string? yes -> return true/false, since division makes no more sense

  if(length_generator > length_bit_string) {

    if(bit_string[0] == '0') return true; // no leading 1 means no rest (see skip) -> finish
    else return false;                    // Rest? yes (and finish)
  } // if

  for(int i = 0; i < length_generator; i++) // bit by bit XOR operation
    if(bit_string[i] != generator[i]) bit_string[i] = '1';
    else bit_string[i] = '0';

  // skip leading zeros

  long skip;

  for(skip = 0; skip < length_bit_string; skip++) {
    if(bit_string[skip] == '1') break;
  } // if

  if((length_bit_string-skip) == 0) return true; // no more rest? yes -> division worked

  return Divisible(&(bit_string[skip]), length_bit_string-skip, generator, length_generator);
} // Divisible
```

Exercise Sensor Networks

Lecture 3: Error recovery and energy efficient MAC

Exercise 3.3: CRC polynomials

(a) Find an easy to identify case in which a given polynomial will fail for a given error.

Solution:

If the error resembles the generator polynomial itself the division will yield not rest. The same is true if this kind of error occurs more than once in the message.

(b) How long does a generator polynomial have to be at least in order to detect every possible bit error if the message has n bits?

Solution:

Theoretically there is no certainty when it comes to error protection because the channel could alter the message and the rest of the division at the same time so that no error is detectable. In general the channel can change every valid code word into another valid code word and there is nothing one can do about it.