Exercise Sensor Networks - (till may 23, 2005)

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.

Exercise Sensor Networks

Lecture 3: Error recovery and energy efficient MAC

Exercise 3.4: Poisson distribution

An audience consists of 10 listeners. Every listener produces an arrive rate of 0.1 phonemes (basic atoms which build spoken language) per time unit. The speaker (in front of the audience) is able to talk at a rate of 2 phonemes per time unit. Each time the speaker encounters 3 or more phonemes the particular time unit is lost and he has to repeat himself. How high is the data rate that can be achieved in this scenario?

Solution:

Average arrival rate per person = 0.1 / for 10 persons = 10 x 0.1

The speaker can handle 0, 1 and 2 arrivals from the audience. These occur with the following probability:

$$P = \sum_{i=0}^{i=2} \frac{(10 \times 0.1)^i}{i!} e^{-10 \times 0.1} \approx 0,92$$

The speaker can talk at 92% of his maximum speed because he is disturbed at 8% of the phonemes.

Exercise Sensor Networks

Lecture 3: Error recovery and energy efficient MAC

Exercise 3.5: Energy efficiency of pure Aloha

A sensor node consumes the following amount of energy:

Basic consumption	: 8 mA
additional consumption for sending	: 20 mA
additional consumption for receiving	: 6 mA

A node must meet a particular energy constraint that requires it not to consume more than 18 mA. How high can the transmission rate per node be chosen in order not to violate the constraint?

Solution:

$(1-e^{-g})\times 20 + e^{-g}\times 6 + 8 <= 18$
$20-20e^{-g} + 6e^{-g} + 8 <= 18$
$e^{-g}(6-20) <= -10$
$e^{-g} >= 10/14$
$1/e^{g} >= 10/14$
$14/10 >= e^{g}$
$\ln(14/10) >= g$
$0.336 >= g$

Note on the expression "arrival rate": The term is misleading in so far as it denotes only the average number of frames (or MAC layer packets) per frame time which are "issued onto the channel". It does however not mean, at least not necessarily, that the packets are actually received by another node.

A node can increase its arrival rate (avg. number of packets it sends to the channel on average) up to 0,336 frames per frame time and will consume less than 18mA.