

# Lecture on Sensor Networks

Copyright (c) 2008 Dr. Thomas Haenselmann (University of Mannheim, Germany).

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

# Communication in sensor networks

## Error control: Cyclic Redundancy Check (CRC)

CRC is based on the idea of polynomial division. Remember:

$$(x^5+x^3+x+1):(x+1) = x^4-x^3+2x^2-2x+3-2 / (x+1)$$

$$-(x^5+x^4)$$

$$\text{-----}$$

$$0 - x^4+x^3$$

$$-(-x^4-x^3)$$

$$\text{-----}$$

$$0 + 2x^3+x$$

$$-(2x^3+2x^2)$$

$$\text{-----}$$

$$0 - 2x^2+x$$

$$-(-2x^2-2x)$$

$$\text{-----}$$

$$0 \quad 3x+1$$

$$-(3x+3)$$

$$\text{-----}$$

$$0 - 2 = \text{Remainder or modulus}$$

Check:  
 $[x^4-x^3+2x^2-2x+3-2/(x+1)]*(x+1) = \dots$

## Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

What's the difference between polynomial division and normal division?

# Communication in sensor networks

## Error control: Cyclic Redundancy Check

A bit string is interpreted as a polynomial by numbering the bits consecutively and, if a bit is set, by adding the corresponding term to the polynomial. In other words: Use the bits as coefficients.

Example:

76543210 position  
 11010101 data bits

$1x^7 + 1x^6 + 0x^5 + x^4 + 0x^3 + 1x^2 + 0x^1 + 1x^0$  is the polynomial corresponding to the given data bits

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

## Error control: Cyclic Redundancy Check

### The principle of CRC:

- Sender and recipient agree upon a „divisor polynomial“, also called generator polynomial.
- Then,  $g$  zeros are added to the message,  $g$  being the degree of the generator polynomial.
- In the next step, the sender divides the message (extended by  $g$  zeros), similar to polynomial division. In most cases there will be a remainder, the result of the division is of no interest.
- The remainder is then subtracted from the message (being extended by the zeros).
- The resulting bit string is now transferred to the recipient. If the message was transmitted correctly, no remainder should emerge on the recipient's side.

### Why?

Because the sender intentionally subtracted the remainder before sending the message. The  $g$  zeros which emerge after the division are interpreted by the recipient as an indication of an error free transmission.

Note: The sender can safely subtract the remainder without harming the message. Why?

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

## Error control: Cyclic Redundancy Check

The only difference to normal polynomial division:

Calculations are binary and after the calculation of each digit a modulo 2 operation is performed!

In other words: Always ignore the carry-over. This simplifies the addition and subtraction significantly.

10101101	01001110
+ 01011100	- 11101010
-----	-----
11110001	10100100

Discovery: Both operations, plus and minus, are equivalent to the XOR operation.

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

## Error control: Cyclic Redundancy Check (CRC)

Example:

Message: 1101011011  
 Generator polynomial: 10011 ( $x^4 + x + 1$ ) = 4<sup>th</sup> degree (5<sup>th</sup> order)  
 Message extended by 4 zeros: 11010110110000

Division: 11010110110000:10011= (the result does not matter)

```

XOR 10011
-----
    010011
XOR -10011
-----
    0000010110 ←
XOR -10011
-----
           0010100
XOR      10011
           001110 = remainder
    
```

**Special note:** The division continues, if the MSB (most significant bit) of the divisor and of the bit string currently being divided is set. Sometimes the bit string has to be extended by new bits (from the message) until the generator polynomial “fits under it”.

```

11010110110000
minus (XOR) 1110
11010110111110 = transmitted message, which should
generate no remainder when divided
    
```

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

Error control: (CRC) – Recognized errors

Which errors are recognized?

For the following analysis separate the error from the message:

Transmitted message (resp. the corresponding polynomial) including an error:  $M(x)$   
 Original message (error free):  $T(x)$

Separate the transmitted message in:  $M(x) = T(x) + E(x)$

with  $E(x)$  being the isolated error. Every bit which is set in  $E$  stands for a toggled bit in  $M$ . A sequence from the first 1 bit to the last 1 bit is called a **burst-error**. A burst-error can occur anywhere in  $E$ .

Question: Does the following division by the generator polynomial  $G(x)$  produce a remainder? If not, we cannot detect the error.

$[T(x) + E(x)]:G(x) =$  „remainder-less“?

$T(x):G(x)$  is divisible without any remainder, because we constructed the message exactly for this property. The analysis is therefore reduced to the question whether  $E(x):G(x)$  erroneously results in no remainder thus passing undetected.

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

Error control: (CRC) – Recognized errors

Which errors are recognized?

1-bit error:

The burst consists of only one error. If the generator polynomial has more than one coefficient,  $E(x)$  with a leading 1 followed by zeros cannot be divided without a remainder. So we are on the safe side with regard to 1 bit errors. Our generator polynomial is at least as good as a parity bit.

Example:

```

1000 (...) 0 : 101 = 1
- 101
----
001 ... continued as above...
```

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation



# Communication in sensor networks

Error control: (CRC) – Recognized errors

Which errors are recognized?

2-bit error:

A 2-bit error must look like this:  $x^i + (\dots) + x^j$ , therefore  $x^j$  can be factored out, which results in  $x^j(x^{i-j} + 1)$ .

It has already been shown that a generator polynomial with more than one term cannot divide the factor  $x^j$ .

- When is a term  $(x^k + 1)$  divided? (with  $k = i - j$ )

For a given generator polynomial this has to be tested for 2-bit bursts with different lengths. Here, the error (inevitably) has the form  $10(\dots)01$ .

What follows is an example program to test whether the generator polynomial

$$x^{15} + x^{14} + 1$$

is useful for detecting 2-bit errors.

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

Error control: (CRC) – Recognized errors

Which errors are recognized?

```

main()
{
  char* generator = "1100000000000001";
  char* bit_string;

  for(int length = 2; length < 60000; length++) {

    if((length % 100) == 0) cout << length << endl;
    bit_string = new char[length+1];

    for(int j = 1; j < length-1; j++) // clear bitstring
      bit_string[j] = '0';

    bit_string[0] = '1'; bit_string[length-1] = '1'; bit_string[length] = 0;

    // test if divisible by generator polynomial
    if(Divisible(bit_string, length, generator, strlen(generator)) == true) {
      cout << "Division successful with length " << length << endl;
      break;
    } // if

    delete[] bit_string; bit_string = NULL;
  } // for

  if(bit_string) delete[] bit_string;
} // main

```

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

**Error control: (CRC) – Recognized errors**

Which errors are recognized?

Error polynomials with an odd number of terms:

**Speculation:** If the generator polynomial contains the term  $(x^1 + x^0)$ , an error string with an odd number of bits cannot be divided.

**Proof by contradiction:** Assuming  $E(x)$  being divisible by  $(x^1 + x^0)$ , the factor can also be extracted:

$$E(x) = (x^1 + x^0) Q(x)$$

So far we only divided polynomials. Now, for the first time we use them as functions and evaluate it for  $x = 1$ .

$(x^1 + x^0)$  equals  $(1 + 1)$  and  $Q(x)$  equals 1, because  $Q(x)$  still contains an odd number of terms (additions are still done modulo 2). Hence follows  $(1 + 1) Q(x) = 0 \times 1 = 0$

But in the beginning we assumed that  $E(x)$  contains an odd number of terms. Thus, the result should have been 1, not 0. As follows, the factor  $(x^1+x^0)$  cannot be extracted. As a consequence,  $E(x)$  is not divisible by  $(x^1 + x^0)$ , if it contains an odd number of terms (or error bits). **Result:** The generator polynomial should contain the term  $(x^1 + x^0)$  to catch all errors with an odd number of bits.

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

Error control: (CRC) – Recognized errors

Which errors are recognized?

Recognition of burst errors of length  $r$ :

The burst error in  $E(x)$  could look like this:  $0001\_anything\_100000$

To move the last bit to the very right, a factor can be factored out:

$E(x) = x^i(x^{(r-1)} + \dots + 1)$ , with  $i$  being the number of zeros on the right side of the last 1.

If the degree of the generator polynomial itself is  $r$  (hence it has  $r + 1$  terms), the error of the form  $(x^{(r-1)} + \dots + 1)$  cannot be divided either, because the generator polynomial is larger than the error to be divided.

**Example (decimal system):** 99:1234567 is not divisible without a remainder (result 0, remainder 99). Detecting burst errors with length  $r$  is trivial in the way that the error itself simply occurs at the end of the division.

Even if the burst is just as large as the generator polynomial (which means  $r+1$  bits), the division yields no remainder only if by chance the error coincides exactly with the generator polynomial. This is possible, but not very likely.

**Example (decimal system):** 1234567:1234567 = 1 (modulo 0)

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

Classic medium access control (MAC) protocols for sensor nodes?

Pure ALOHA

Idea: Everybody may transmit whenever desired, but only a frame of max. length. Other participants who are already sending are not regarded.

Collision:

If two frames overlap, both of them are considered as destroyed. But both participants are able to detect the collision and they can send their message (frame) once again. Of course both frames would collide again, hence there would never be a valid transmission.

Solution:

Every sender waits for a random amount of time before starting to transmit (in the case of sensor nodes this time could be „overslept“ to save on energy). The solution to the collision problem is that the transmission of the participant with the shorter delay has (hopefully) already finished before the one with the longer delay starts to send. In the worst case another collision will happen and the process has to be repeated.

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

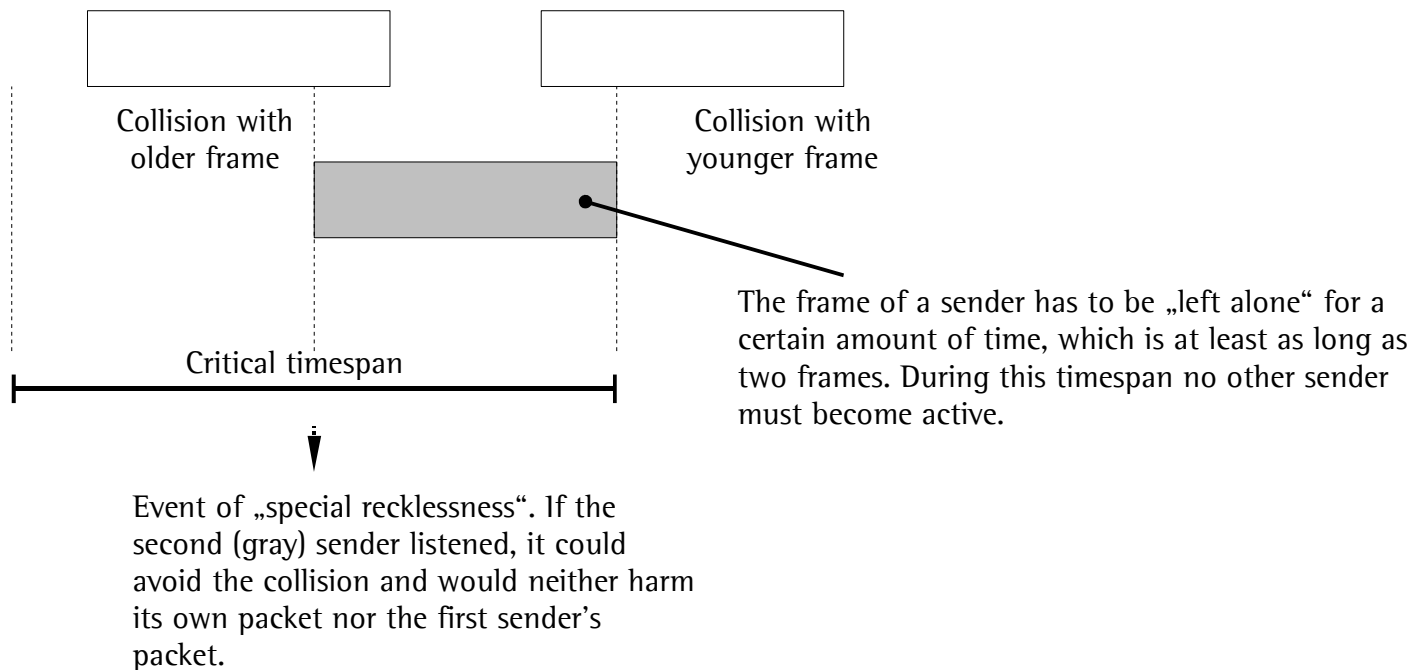
Simulation

# Communication in sensor networks

Classic medium access control (MAC) protocols for sensor nodes?

Pure ALOHA

The frame can collide on both ends:



Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

Classic medium access control (MAC) protocols for sensor nodes?

Pure ALOHA: Analytical review

Arrival rate  $G$

Average number of transmission attempts per frame length by a sender.  $G = 0.5$  means for instance, that on an average every second idle frame on air is occupied by the sender (with or without collision).

Transmission rate  $S$

Number of frames that actually reach their destination.

$S = 0$ : no frame reaches its destination because...

- a) no frames are sent
- b) so many frames are sent that all of them collide

$S = 1$ : every frame can be transmitted without collision, e. g. because there is only one sender

$P_s$

Probability that frames are transmitted successfully

$S = GP_s$

Apparently there are two extremes:

- a) The senders create hardly any arrivals ( $G=0$ ). This results in good conditions for the transmission, because the medium is always free ( $P=1$ ). On the other hand, no one uses it.
- b) The senders create many arrivals ( $G$  large). This results in continuous collisions ( $P=0$ ), hence nothing is transmitted without collision. The optimum lies somewhere in between.

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

Classic medium access control (MAC) protocols for sensor nodes?

Pure ALOHA: The Poisson distribution

Lambda is the arrival rate per time unit. The probability for n arrivals per time unit is calculated like this:

$$P_{\lambda}(n) = \frac{\lambda^n}{n!} e^{-\lambda}$$

lambda = number of arrivals per time interval

1/lambda = mean time between two consecutive arrivals

n = number of arrivals, for which we want to obtain the probability P(n)

Which n will yield the highest probability?

Remark: The use of the Poisson distribution requires an exponential distribution of the time of arrival between two consecutive events. One can usually assume this in case of mutually independent results and a large number of participants.

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation



# Communication in sensor networks

Classic medium access control (MAC) protocols for sensor nodes?

Pure ALOHA: Analytical review

**Example:** We know that a department in a store registers an average number of two arrivals of customers per minute during lunchtime. What are the probabilities for 0, 1, 2 etc. customers, resp. that at least 0, 1, 2, etc. customers arrive?

$\lambda = 2, n=0,1,2, \dots$

n	P(n)	Sum
0	0,1350	0,135
1	0,2710	0,406
2	0,2710	0,677
3	0,1800	0,857
4	0,0900	0,947
5	0,0360	0,983
6	0,0120	0,995
7	0,0030	0,998

One salesperson has nothing to do with a P' of 13%.

With P'=40%, one salesperson is enough for customer care.

With P'=59% there's a need for at most 2 salespersons.

In 32% of the cases at least a third salesperson would be necessary.

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

Classic medium access control (MAC) protocols for sensor nodes?

Pure Aloha: Analytical review

As seen before, a successfully transmitted packet needs at least two frames. In other words there have to be exactly 0 arrivals of other packets during two periods.

$$P_s(0) = \frac{(2gN)^0}{0!} e^{-2gN}$$

The diagram illustrates the components of the equation  $P_s(0) = \frac{(2gN)^0}{0!} e^{-2gN}$  with arrows pointing to their respective parts:

- Aloha-typical 2 slot problem**: Points to the entire equation.
- Arrival rate per participant (transmission attempts)**: Points to the variable  $g$ .
- Number of participants**: Points to the variable  $N$ .
- Number of desired arrivals**: Points to the exponent  $0$  in the numerator  $(2gN)^0$ .

How does the probability  $P_s$  of the chance to send successfully vary with the number of participants?

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

Classic medium access control (MAC) protocols for sensor nodes?


Pure ALOHA: Analytical review


Packet throughput of pure Aloha

On the last page we calculated the probability of the channel being idle when there is a certain average number of packets to be sent (the arrival rate).

In fact, the achievable data throughput  $L$  (having  $n$  participants with a transmission rate of  $g$  packets per frame length and participant) is more interesting.

$$L(N, g) = gN \frac{(2gN)^0}{0!} e^{-2gN}$$


 Arrival rate (in packets per frame length) on the channel
 


 Probability of success

What's the maximum packet throughput which can be achieved with classic ALOHA?

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

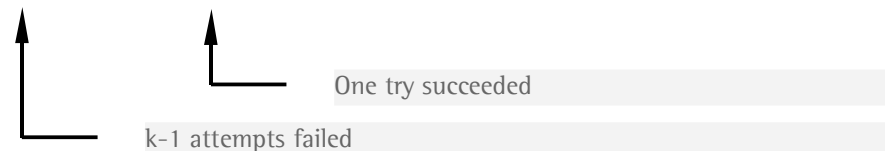
# Communication in sensor networks

Classic medium access control (MAC) protocols for sensor nodes?

Pure ALOHA: Analytical review

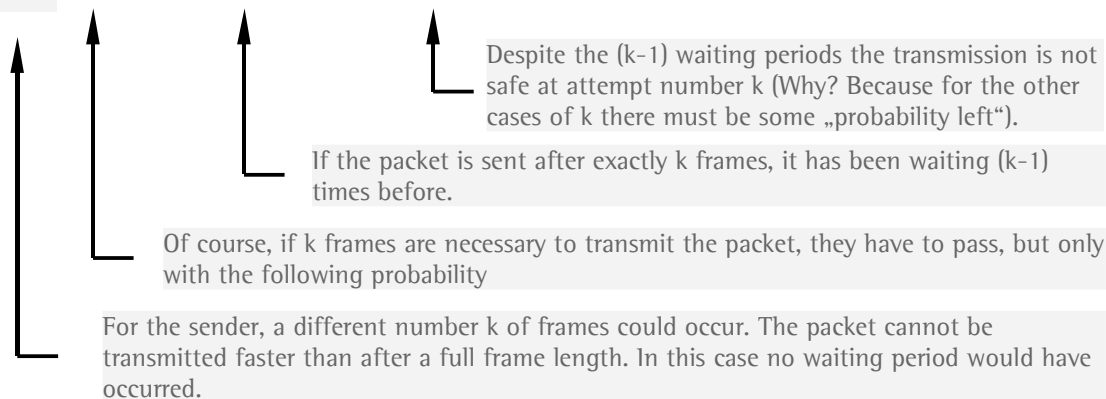
Probability of successful transmission on the  $k^{\text{th}}$  attempt to send:

$$P(k = K) = (1 - P_S)^{k-1} P_S$$



Average time to deliver one packet:

$$D = \sum_{k=1}^{\infty} k (1 - P_S)^{k-1} P_S$$



Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

Classic medium access control (MAC) protocols for sensor nodes?

Energy efficiency of pure Aloha

Whenever there is nothing to send, the energy consumption alternates only between  $P_{TX}$  and  $P_{RX}$ , because the medium is always monitored in Aloha.

$$b_1 = 1 - \frac{(1g)^0}{0!} e^{-1g}$$

Rate only created by the sender

The sender itself produces 0 arrivals (is idle)

Complementary case, which means the sender is busy

$P_{RX}$  Energy consumption during Reception  
 $P_{TX}$  Energy consumption during Transmission

$$P^{RA} = b_1 P_{TX} + (1 - b_1) P_{RX}$$

Transmission energy

Reception energy  
 Remark: This includes inefficient idle-listening, too.

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

Classic medium access control (MAC) protocols for sensor nodes?

Energy efficiency of classic Aloha

Example

$$\begin{aligned} P_{\text{base consumption}} &= 8\text{mA} \\ P_{\text{TX}} &= 20\text{mA} \\ P_{\text{RX}} &= 6\text{mA} \end{aligned}$$

$$b_1 = 1 - \frac{(1g)^0}{0!} e^{-1g}$$

$$P^{\text{RA}} = b_1 P_{\text{TX}} + (1 - b_1) P_{\text{RX}}$$

A frame length is 10ms, i. e. 100 frames/second.  
Nodes will send once per second.

$$G=0.01 \Rightarrow b_1 = \sim(1-0.99) = 0.01$$

$$P^{\text{RA}} = 0.01 \times 20 + (1-0.01) \times 6 + 8 = \sim 14.14\text{mA}$$

Battery with 2,000mAh will last for  $2,000 \times 60 \times 60 / 14.14 = \sim 509,193\text{s} = \sim 141$  hours

**Result:** Hardly 6 days of operation do not justify to deploy a network for many applications.

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

Example for medium access control (MAC) protocols:

Is Pure Aloha suitable for sensor networks?

- + A sender can send anytime, i. e. whenever the need to send emerges.
- + If a transmission fails, the randomly determined waiting period has to pass, during which the node can switch to sleep mode.
- A node has to listen to the channel all the time since messages can also occur at any time. Especially for long-distance routing, a node may not know in advance when messages have to be received and forwarded, because in this case, the node is not the initiator itself, but functions as the router.

Permanent readiness to receive is not an option for sensor nodes. The problem has to be solved more energy efficiently.

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

**Energy efficiency of Aloha**

Simulation

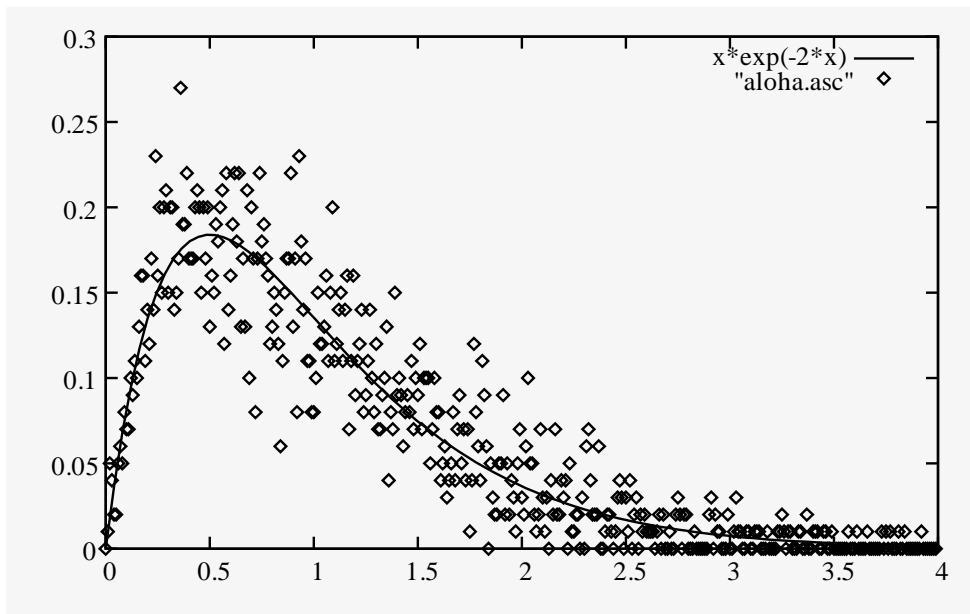
# Communication in sensor networks

Classic medium access control (MAC) protocols for sensor nodes?

Simulation of different MAC-protocols

After the analytical view the Aloha procedure is now simulated using a small program.

**Given:** 100 stations; 100 frame lengths, each divided into 100 simulation time units



**Observation:** The simulation adapts to the curve well. Why do clearly higher or lower data throughputs still occur in some simulation runs in practice ?

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation



# Communication in sensor networks

Classic medium access control (MAC) protocols for sensor nodes?

Simulation of different MAC-protocols

```

const long MAX_TIME      = 10000L;
const long NO_STATIONS  = 100L;
const long FRAME_LENGTH = 100L;
    long medium_occupied_till = -1;
    long survival_timer     = -1;
    long successful_packets = 0;
    long no_frames          = MAX_TIME / FRAME_LENGTH;

class Station
{
public:
    void Init() {};
    void TriggerSend(long);
}; // class Station

void Station::TriggerSend(long current_time)
{
    if(current_time < medium_occupied_till) {
        survival_timer = -1;
    } // if
    else survival_timer = 0;

    medium_occupied_till = current_time + FRAME_LENGTH;
} // Station::TriggerSend
    
```

Cyclic Redundancy Check

CRC analysis

Aloha for sensor networks

Energy efficiency of Aloha

Simulation

# Communication in sensor networks

Classic medium access control (MAC) protocols for sensor nodes?

Simulation of different MAC-protocols

```

main()
{
    Station* station    = new Station[NO_STATIONS];

    for(long i = 0; i < NO_STATIONS; i++)
        station[i].Init();
    for(long arrival_rate = 0; arrival_rate < 400; arrival_rate += 1) {
        medium_occupied_till = -1; survival_timer = -1; successful_packets = 0;
        for(long time = 0; time < MAX_TIME; time++) {
            for(long station_index = 0; station_index < NO_STATIONS; station_index++)
                if((abs(rand()) % (100*NO_STATIONS*FRAME_LENGTH)) < arrival_rate)
                    station[station_index].TriggerSend(time);

            if(survival_timer != -1) survival_timer++;
            if(survival_timer == FRAME_LENGTH) {
                successful_packets++;
                survival_timer = -1;
            } // if
        } // for

        double overall_arrival_rate = ((double)arrival_rate)/100.0;
        cout << overall_arrival_rate << " " <<
        (double)successful_packets/(double)no_frames << endl;
    } // for

    delete[] station;
} // main
    
```

Cyclic Redun-  
dancy Check

CRC analysis

Aloha for sensor  
networks

Energy efficiency  
of Aloha

Simulation