UNIVERSITÀ DI PISA

**Computer Engineering**
**Performance Evaluation of Computer Systems and Networks**

# Evaluation of a Remote Sensing System

TEAM MEMBERS:

Francesco Taverna

Gabriele Pianigiani

Saverio Mosti

Academic Year 2023/2024

# Contents

# Introduction

**Remote sensing** is the acquisition of information about an object or phenomenon without making physical contact with the object in question.

## Problem Description

A remote sensing system is composed of $N$ **sensing nodes** that move randomly within a 2D floorplan of size $L \times H$, they move from a **waypoint** to another.

A waypoint is defined by a pair of coordinates $(x,y)$ that are random variables and a speed $s$. Each sensing node when arrives in a waypoint **a**, selects a new waypoint **b** and moves towards it in a selected constant speed.

In the same 2D floorplan there are $M$ **access points** at fixed random positions. Access points are deployed according to a uniform distribution and are connected to one remote **sink node**.

Sensing nodes are equipped with a wireless interface and tries to deliver messages to the sink node through the access points: every $T$ seconds each sensing node sends a broadcast message, which is successfully received with probability $P_{succ}$ by access points located within the sensing node's transmission range $D$. Messages that are successfully received by the access points are then forwarded to the sink node, once the access point sends the message it surely arrives to the sink node.

Evaluate at least the overall rate of correctly delivered messages per second for various values of $M$, $D$ and $P_{succ}$.

At least the following scenario has to be evaluated: uniform distribution of $x$ and $y$.

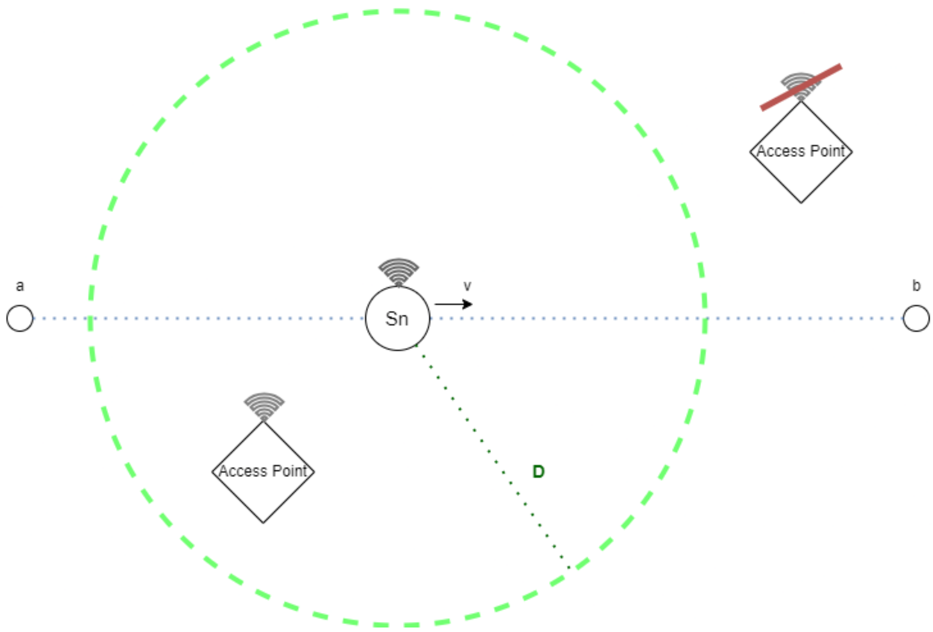


*Figure 1 Remote sensing system scenario*

## Objectives

The objective of the project is testing the effectiveness and efficiency of a series of different scenarios obtained by varying determined parameters such as the dimension of the working plan, the quality of the receiver and the transmission range of the transmitter. In particular for "effectiveness" we mean the maximization of the number of unique packets correctly sent to the sink node per unit of time and the "efficiency" regards the minimization of the number of duplicates per message. We want to find out what are the parameters that changing have the most influence on the number of successful messages and the number of duplicates per message, moreover, will be analysed the best "trade-off" configuration of most important parameters.

## Performance indexes

To assess the performance of the system we took as a reference the following indexes:

- The mean number of non-duplicates messages successfully received E[N_unique_succesful]: defined as the mean number of messages that are not duplicates and successfully arrived to the sink node.
- The mean number of duplicates E[N_duplicates] per message: defined as the mean number of duplicates generated per message.

# Modelling

## General assumptions

In our case study we made the following assumptions:

- Transmission delay between a sensing node and an access point is considered null.
- Probability that two different packets generated by two different sensing nodes are received simultaneously by an access point is negligible, thus that the model which represents the access point has no queueing.
- Sink node's model has no queueing because it is not within the scope of our simulation campaigns.
- All the distributions used are uniform.

## Preliminary validation

Before the implementation, a preliminary validation is necessary to verify that the model is correct. In order to do so, the previous general assumptions should be analysed.

- Transmission delay between a sensing node and an access point can be considered null because we assume that a sensing node moves with a speed of the order of tens km/h, so the transmission delay of a packet (regardless of the number of bits) is negligible.
- We consider null the probability of having a collision at the access point because the above motivations and in general it is very unlikely the case in which two sensing nodes transmit simultaneously (and in a synchronized way) to the same access point. Furthermore, we don't care about the delay.
- The model of sink node has no queueing because we are not interested in modelling it, we are interested in modelling other actors in the system, it just has the role of storing messages.

## Factors

The following factors may affect the performance of the system:

- **M**: number of access points in the system;
- **N**: number of sensing nodes in the system;

- **$P_{succ}$**: probability of success in the transmission of sensing nodes;
- **Access_point_position**: access points can be positioned according to uniform random distribution in whole working plan or in a single slot (*in this case the working plan is divided in M squares*) or in a deterministic way (*still divided in M squares but each access point is in the middle of the square*);
- **H, L**: Dimension of working plan;
- **$V_{max}$**: Maximum possible speed of sensing nodes;
- **T**: Transmission period of sensing nodes;
- **D**: Transmission range of sensing nodes;

# Implementation

To implement the system some modules and a new message type have been defined.

## Modules

All the following modules are defined inside the **RemoteSensingSystem** network:

- **Sensing Node**: simple module which represents a device that moves in the working plan and sends messages in broadcast;
- **Switch**: simple module which receives messages from all the sensing nodes and forwards them to the access points;
- **Access Point**: simple module which represents a static device which forwards messages to the sink node;
- **Sink Node**: simple module which receives messages from the access points.

## Modules' behavior

### Sensing Node

A sensing node starts in a random position within the working plan, if the speed is not null, then generates, according to the chosen distribution, the new coordinates of the next waypoint and the speed at which the node will travel to that waypoint. During the transmission it sends in broadcast a message every T second, if the speed is null, it will send messages every T second without moving, the message sent will reach the switch. In our simulations the speed is uniform randomly generated between an upper and lower bound.
The lower and upper bounds are set to zero only in one degeneracy tests, in the other case are greater than 0.

### Switch

A switch is a module that generates the positions of each access point inside the working plan. If the *Access_point_position* is set to "random" the positions are generated according to uniform distribution, otherwise it splits the working plan in squares and places an access point inside each square, if the parameter is set to "deterministic" the access point will be placed in the centre of the square, if it is "variable", it will be placed in a uniform random position inside the square.

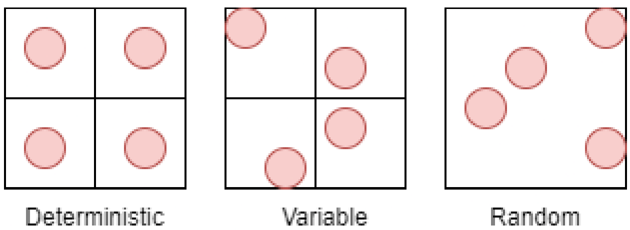


*Figure 2 Different kind of access point distribution*

It receives messages from all the sensing nodes, once it receives a message, compares the range of that sensing node with the distance between the sensing node and all the access points, if no access points are inside the range, the switch drops the message, otherwise sends it to the access points inside the range.

### Access Point

An access point is a simple module, at the beginning it receives messages from the switch, then computes with uniform distribution a probability ($P_{succ}$) and according to it sends the message to the sink node, otherwise it is dropped.

### Sink Node

A sink node is another simple module, it only receives messages from the access points and computes the number of unique received messages and duplicates.

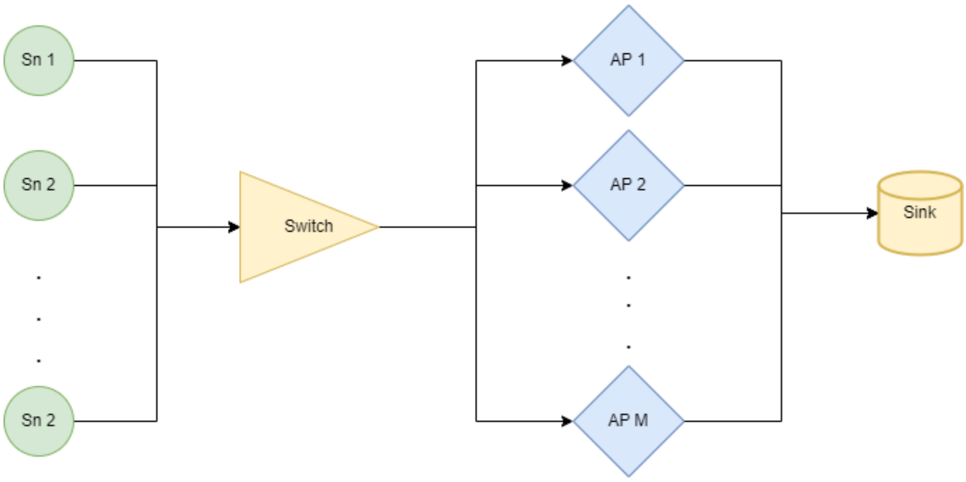


*Figure 3 Structure of the Remote Sensing System scenario in OMNeT++*

### Message

To accurately represent the packets in the system, a new type of message, named "**Remote_Sensing_System_Message**", has been defined as a subclass of *cMessage*. In this new format, we added four fields, the coordinates x and y of the sensing node, its range D and the id of the sensing node that have sent the message.

## Verification

In this section we analyse the implementation's behaviour to check its coherence also with respect to the theoretical model.

We performed three different verifications:

1. Degeneracy
2. Consistency
3. Continuity

And then we tested the theoretical model also with analytical computation.

All the verification experiments have been run using the following parameters:

- Repetitions for degeneracy, consistency, continuity: **1**;
- Simulation time for degeneracy: **202.5s**;
- Simulation time for consistency and continuity: **802.5s**;
- Working plan dimensions (H x L) except the first three degeneracy: **1000m x 1000m**;
- Maximum speed of sensing nodes except the first degeneracy: **5 m/s**;

- Minimum speed of sensing nodes except the first degeneracy: **1 m/s**;
- Maximum distance for sensing nodes: **10m**;
- Minimum distance for sensing nodes: **1m**.

## Degeneracy Tests

The degeneracy verification is used to analyse the system under specific extreme conditions, by degenerating some parameters to see if, even in this case, it behaves as expected.
We observed the following behaviours:

- **Vmax=0,** each sensing node is static, it doesn't move as expected;
- **N=0**, if we don't have any sensing node, we can see that all the metrics computed are null as expected;
- **M=0**, if we don't have any access point, we can see that no messages arrive to the sink node, there aren't any duplicates, there aren't any failed receptions due to $P_{succ}$ and the number of messages sent to nobody is equal to the total number of messages sent as expected;
- **$P_{succ}$ = 1**, if the probability of success is 1, the number of failed receptions is null as expected;
- **$P_{succ}$ = 0**, if the probability of success is 0, the number of successful messages is equal to zero and the number of failed messages is the difference between total messages and messages sent to nobody as expected;
- **D=0**, if the transmission range of sensing nodes is null, we have that the number of successful messages, number of duplicates and number of failed receptions are 0 because the number of messages sent to nobody is equal to the total number of messages sent as expected;
- **D=2*H**, if the transmission range is the maximum possible, the number of messages sent to nobody is 0 and the duplicates are 15 times the number of total messages sent by the access points in case there are 16 access points because each sensing node reach all the access points.

## Consistency Tests

The consistency verification was carried out to check whether the system reacts in a consistent way or not.
We observed the following behaviours:

- **T=1s**, if the period of transmission is 1s instead of 2s, we observe that the number of messages sent is doubled as expected;
- **T=4s**, if the period of transmission is 4s instead of 2s, we observe that the number of messages sent is halved as expected;

## Continuity Tests

The continuity test checks the correctness of the model, it also proves if a slight variation of the input will produce analogous changes in the output. We observed the following behaviours:

- **$P_{succ}$ = 0.7**, in this case the mean number of failed messages is 3127.2;
- **$P_{succ}$ = 0.65**, in this case the mean number of failed messages is bigger than the previous case, 3652.95 as expected;
- **$P_{succ}$ = 0.75**, in this case the mean number of failed messages is smaller than the first case than the previous case, 2851.5 as expected;
- **D = 80m**, in this case the mean number of unique messages successfully arrived is 17596.05 and the mean number of total duplicates is 68;
- **D = 90m**, in this case the mean number of unique messages successfully arrived is 22436.2 which is bigger than the previous one and the mean number of total duplicates is 123.7 which is bigger than before, as expected;

- **D = 70m**, in this case the mean number of unique messages successfully arrived is 14241.5 which is smaller than the first case and the mean number of total duplicates is 18 which is smaller than the first case, as expected.

## Theoretical verification

Theoretical model verification was performed to see if the results we obtained from the implementation of the system are the same as we obtain analytically from that model. For this verification we have created a simulation scenario in which access points are fixed at the centre of each square in which the working plan is divided into, and each sensing node doesn't move. We have fixed elements in order to simplify calculations otherwise with random positions and movements was difficult to model analytically the system. In this scenario we have 20m for H and L parameters, the work plan is 400m$^2$, 4 access points and 5 sensing nodes, 0.8 for $P_{succ}$ and T is equal to 1s, the range of each sensing node (D) is 4m and the simulation time is 200s.

Each access point has these fixed coordinates: (5;5), (15;5), (5;15), (15;15).

Each sensing node has these coordinates that are randomly computed: (10.97;11.85), (14.30;16.88), (12.05;17.15), (10.89;16.94), (8.47;12.47).

Running the simulation with these values we get the following results:

| Parameter | Value |
|---|---|
| Number_of_duplicate_messages_successfully_arrived_Stat | 0 |
| Number_of_duplicates_Stat | 0 |
| Number_of_failed_receptions_Stat | 78 |
| Number_of_Messages_sent_to_nobody_stat | 600 |
| Number_of_sent_messages_Stat | 1000 |
| Number_of_Successful_Messages_Stat | 322 |
| Number_of_unique_messages_successfully_arrived_Stat | 322 |

There are 5 sensing nodes, every one second each node sends a message, the duration of the simulation is 200 seconds, so the number of messages sent is $5 \cdot 200/1 = 1000$. Computing the distances between access points and sensing nodes using the Euclidean distance only (12.05;17.15) and (14.30;16.88) have the range which includes one access point, so there are no duplicates.

Other three sensing nodes don't have any access points inside their ranges, so the number of messages sent to nobody in 200 seconds is $3 \cdot 200/1 = 600$.

The number of messages sent with success by the two sensing nodes, considering $P_{succ} = 0.8$, is $2 \cdot 200/1 \cdot 0.8 = 320$, while the number of messages failed is about $2 \cdot 200/1 \cdot 0.2 = 80$. The number of unique messages successfully arrived at the sink node is the same of the total number of successful messages because there are no duplicates.

All the analytical values are coherent with the values obtained with the simulation, only the number of failed receptions and successful messages is a bit different because of the computation of the probability $P_{succ}$.

## Calibration

The calibration phase is necessary to set all the configuration variables, both the factors and configuration parameters that affect the performance of the system.

About the factors of the system, we want to obtain meaningful values like those of a real system but scaled down for software issues in the simulation tool.

To find concrete values, we did some research, and we got the following results:

- The average transmission range of wireless devices such as smartphones is about 100 meters[1].
- The average period of transmission, used as a reference, for each sensing node is about 104.5ms (beacon transmission period of an access point), for the simulations we have used 200ms in order to have less messages sent[2].
- The maximum speed of sensing nodes is 5 m/s which is 18 km/h, and it is equal to the average speed reached by a normal bicycle[3].

For variable factors we have selected these ranges:

| Factor | Interval | Unit of measurement |
|---|---|---|
| Maximum Speed (Vmax) | [4;10] | m/s |
| H (L=H) | [1000;11000] | m |
| D | [50;170] | m |
| $P_{succ}$ | [0.3;0.8] | probability |

We also need to calibrate the warm-up time and the simulation time. The former is important because we want to study the system when the transient is terminated, during the steady state when we get more unbiased statistics. Instead, the second is relevant because too long simulation time is unnecessary to get meaningful data, since after a while no new information is acquired and the results are too heavy in terms of memory occupied.

## Warm-up time

To assess a reasonable warm-up time, we simulated and examined the evolution of performance indices of our model in the first part of its advancement, trying to observe from what point onwards they start to stabilize.

We have evaluated the number of messages received by the sink node every 60 seconds, we have done 5 repetitions and using Excel we got the result in figure 3.

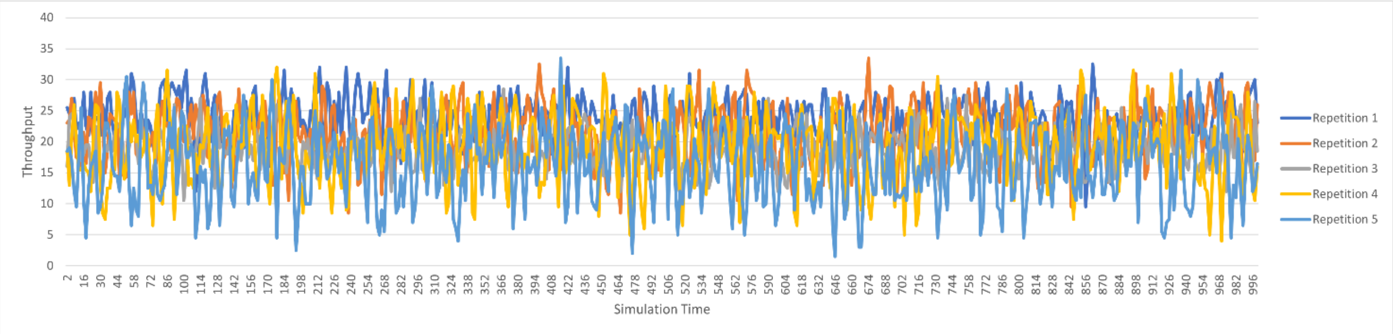


*Figure 4 Warm-up calibration: The graph show how the number of successful received messages change over time.*

[1] https://www.ninjamarketing.it/2011/12/29/tethering-wi-fi-converti-il-tuo-smartphone-in-un-router-how-to/
[2] https://mentor.ieee.org/802.11/documents
[3] https://www.nike.com/it/a/quanto-tempo-e-necessario-per-percorrere-un-chilometro-a-piedi
[3] https://www.uniroma1.it/it/pagina/bici-scendere-dallauto-si-puo#:~:text=In%20bicicletta%20la%20velocit%C3%A0%20media,km%2Fh%20per%20le%20moto.

Observing the graphic, we can see that it is stable since the beginning, although nothing changes in terms of rates because from the beginning it is in the canonical interval, in 2 seconds each device sends about 10 messages, so the rates are still consistent.
We have chosen 2.5 seconds as the warm-up period in order to allow the preliminary computations done by the switch, such as assign coordinates to access points, because we don't know the computational power of the machine on which the simulation tool will be used. In our case, having used not too high values, the computations are done very fast, but if some messages were sent to the switch before the end of computations, the results would be inconsistent.

## Simulation time

Inside the system messages are sent every 200ms, thus we have decided a simulation time of 30 minutes (*1800s*), in this time interval we have 7.5 million packets as an estimate (due to the high number of sensing nodes and a little T): they are more than enough to analyse the system. Moreover, the system is stable from the beginning and significant data are immediately available. By carrying out multiple tests with different durations we can observe that only the amount of data changes.

# Experiments

In this section will be conducted experiments on our validated and verified model. We are going to analyse how the main factors variation influences the number of unique messages correctly arrived at the sink node per unit of time and the average number of duplicates generated per message. From the number of unique messages correctly arrived can be computed the main index E[N_unique_successful], by dividing for simulation time.

We have chosen the duplicates index in our experiments because, as well as studying the rate of successful messages changing the factors and finding the best configuration to maximize it, it's interesting to analyse also how the first purpose leads to messages duplication.

Therefore, this study is aimed at finding a trade-off between the first and second index: if the number of duplicates goes out of control (*it's very likely due to the maximization of successful messages*) the sink node has to manage a big load, hence it needs a high computational power and a big memory to store all the messages.

Maximizing the first causes a high increment of the second: the main accountable of this phenomenon is the factor M which can be described as "*coverage*", in other words is the number of access points per unit of space (it's a sort of access point's density).

Increasing the coverage will decrease the number of messages sent to nobody, hence increasing the number of messages that reach at least an access point. For this reason, the number of duplicates increases because it's more likely that a message is sent to two or more access points, instead of only one.

In our simulation we didn't change M but rather H (*equal to L, it's a squared plan*) because our objective is to modify the access point's density that will affect the coverage, thus also the analysed indexes. This choice was made to simplify the coding: indeed, in order to guarantee different seeds for each random generator of coordinates and $P_{succ}$, we couldn't change the number of access points.

## 2kr – Average number of unique successful messages per unit of time

We started with 2kr approach to understand which factors between D, H, Vmax and $P_{succ}$ influence mostly the index of unique successful messages per unit of time but also the effect of their interplay. First of all, the normal distribution of residuals has to be checked, it can be done with the following QQ plot:

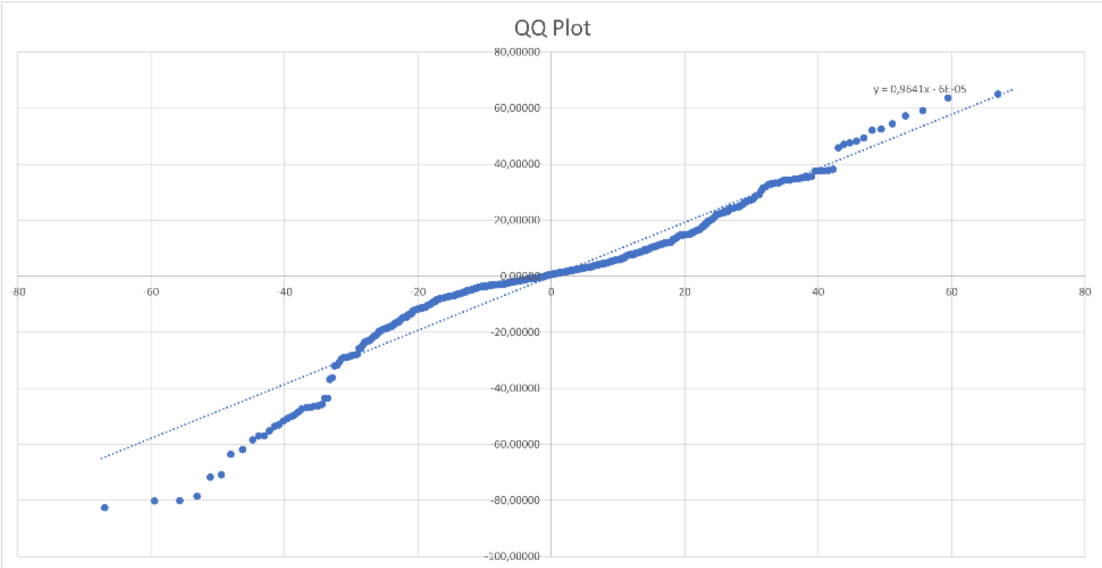


*Figure 5 QQ Plot of the number of unique successful messages arrived at the sink node per unit of time.*

The sample quantiles approximately follow the quantiles of a normal distribution, because the points are more or less on the tendency line. It can be seen that on the bottom left some points tends to stray from the line, but the last points of that little tail are back to the linear trend: we suppose that incrementing the number of repetitions this tail would be closer to the straight line. Thus, the normal hypothesis can be assumed approximately and sufficiently verified. The coefficient of determination $R^2$ is equal to 0.99918 so it's very close to 1.

### The Homoskedasticity Test

Then we computed the homoskedasticity to verify that the standard deviation is constant and can be observed that the following plot does not show a particular trend between point of the same color. This hypothesis is also confirmed by the fact that the residuals are two magnitude orders less than the mean.

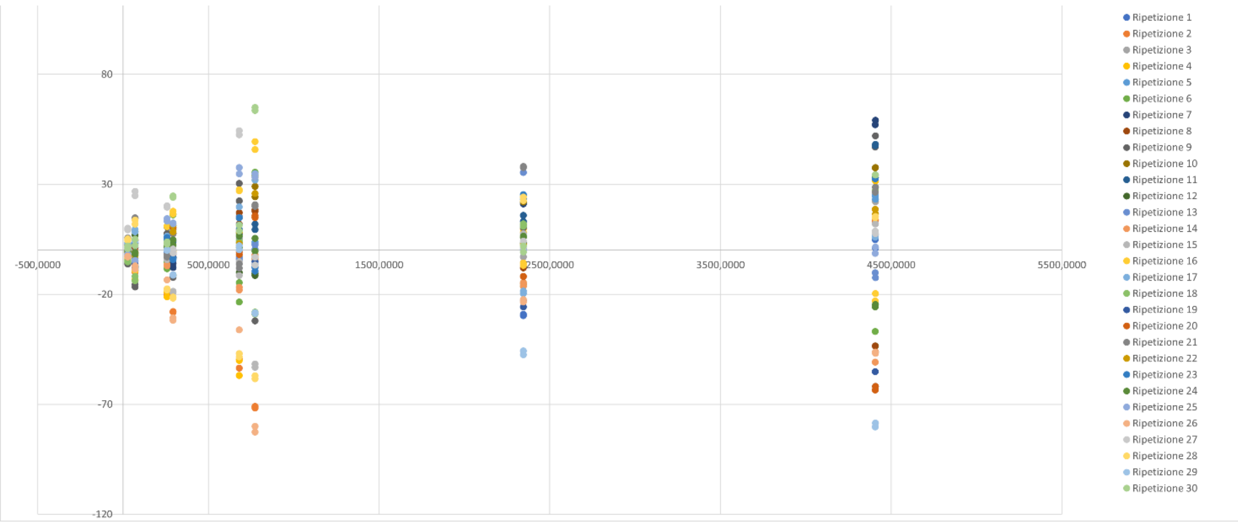

*Figure 6 Homoskedasticity test*

### The 2kr Conclusions

The confidence intervals of the effects with 99.9% in the 2kr don't include 0, so they can be assumed significant at the specified level of confidence.

| Factor | Variation |
|---|---|
| H | 32.55% |
| D | 35.16% |
| Psucc | 6.92% |
| H + D | 17.95% |
| H + Psucc | 2.93% |
| D + Psucc | 3.29% |
| H + D + Psucc | 1.11% |
| Others | < 1% |

This percentages confirms that the most important factors are H, D, and their interplay, indeed the sum of these three is more than 85%.

Moreover, the probability of success can't be considered negligible indeed it is around 7%, obviously much less than the previous ones. These results are not surprising because thinking about the success of communications, it's clear that the coverage (*access point's density*) is the main factor: more the plan is full of access points, more likely is that a sensing node reaches at least one access point.

## Full factorial – Average number of duplicates per message

For the analysis of the duplicates index, initially we performed a 2kr analysis, but the QQ plot does not show linear trend when comparing the residual and normal distributions.
Afterwards we did a full factorial with 30 repetitions and considering just H and D for two main reasons:

- $P_{succ}$ doesn't influence the number of duplicates because switch counts duplicates before the probability check done by the access points. When the transmission range includes two access points a duplicate is generated independently of $P_{succ}$.
- $V_{max}$ is irrelevant in both indexes since we had performed a lot of simulations changing it and nothing happened in the results, unless the degeneracy case in which it is equal to 0.

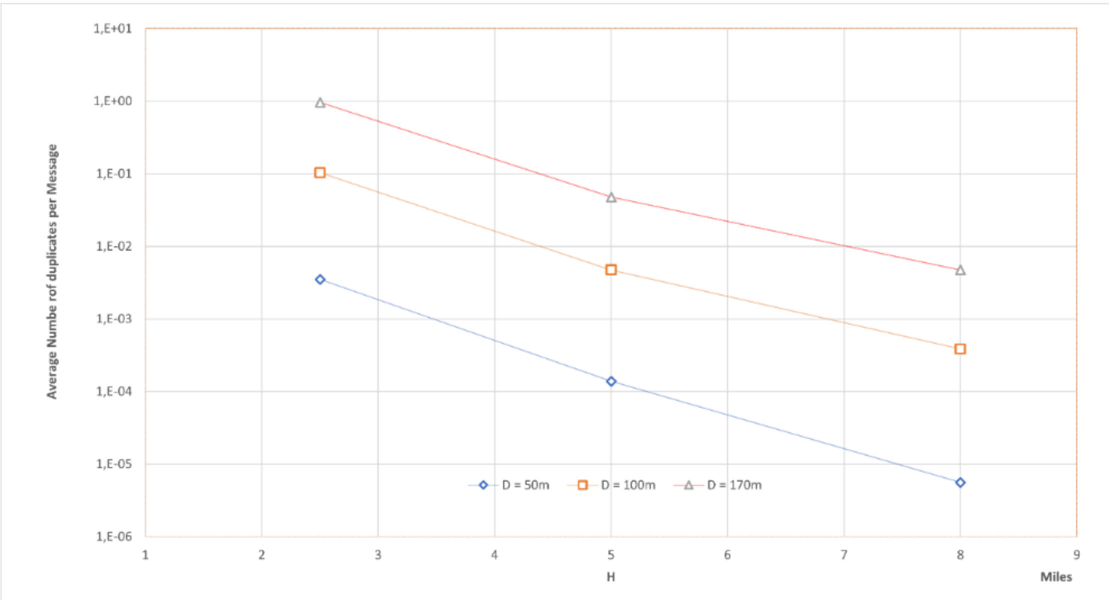


*Figure 7 The effect of the interplay between H and D on the average number of generated duplicates per message.*

In Fig7 is shown the variation of the average number of duplicates per Message by varying the transmission range for each working plan dimension took into consideration. On the y axis, where it's represented the index, the numbers are expressed in logarithmic scale due to the fact that the values assumed are below 1 and in the last cases tends to 0 (*very low access point's density*). The plotted segments seem to be parallel and thus independent, in fact they are not because this trend is due to y axis scale. Considering the maximum value of D (170m) and a big plan as the last case (*H = 8000m*), the variation of the index is very low, and the real slope of the lines is almost flat, so it's important to refer to the values indicated that are almost hundred times less than a normal value as 1 duplicated per message.

For each value of D, it's clear that the index decreases of 2 orders of magnitude tending to 0 when the plan is very big, hence H is the main factor that influences duplication. In the meantime D it's also relevant because comparing the lines with different D, the index is 1 order of magnitude lower.

# Conclusions

The objective of this work was to find a good trade-off between the effectiveness (*maximize the average number of unique messages received by the sink node*) and the efficiency of this system (*minimizing the average number of duplicates*).
The best configuration obviously depends on the priorities of the system and of its possibilities:
For maximizing the effectiveness, the best thing to do is to increment the "access point density", so add more access points. This would be a perfect solution only if the sink node has an efficient system for identify a duplicate (*the sink has already received a message like that*) and a sufficient memory.

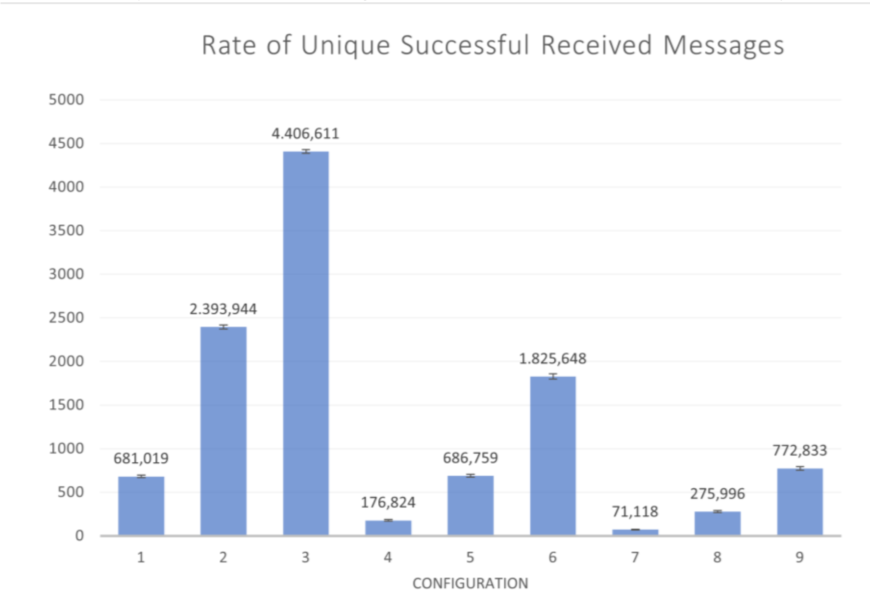


*Figure 8 Successful messages index comparison for different couples of factors D and H.*

| Configuration | D | H |
|---|---|---|
| 1 | 50m | 2500m |
| 2 | 100m | 2500m |
| 3 | 170m | 2500m |
| 4 | 50m | 5000m |
| 5 | 100m | 5000m |
| 6 | 170m | 5000m |
| 7 | 50m | 10000m |
| 8 | 100m | 10000m |
| 9 | 170m | 10000m |

Moreover, as we can see in Fig9 the average number of duplicates increases exponentially with D (*if H is constant*) then the traffic inside the system would increase exponentially (*In Fig9 the results are expressed in a logarithmic scale, for this reason they appear "mirrored"*).

An excessive number of duplicates can generate:

- Delays on the access points.
- Excessive utilization of the sink node.
- More power consumption.

So, the important thing to consider is to <u>not increment too much the density of the access points</u>, because the number of duplicates can grow exponentially with it.

But have duplicates can be an advantage too: the access point can fail the reception of a message, in that case, if a message has been duplicated has more chances to arrive to the sink node. A good average number of duplicates per message is around 1, because every message has been duplicated more or less 1 time and the number of messages that are circulating in the system is doubled (*so it is still contained*).

In conclusion, in this analysis we have evaluated that the best configuration is the third one, in which D is equal to 170m and H is equal to 2500, i.e. the one with lower plan area and maximum transmission range. This is just a specific analysis in which have been considered few values for the factors and maybe a more detailed one can represent better the previous inferences, for example the increment of the number of access points can lead to an higher rate of duplicates.

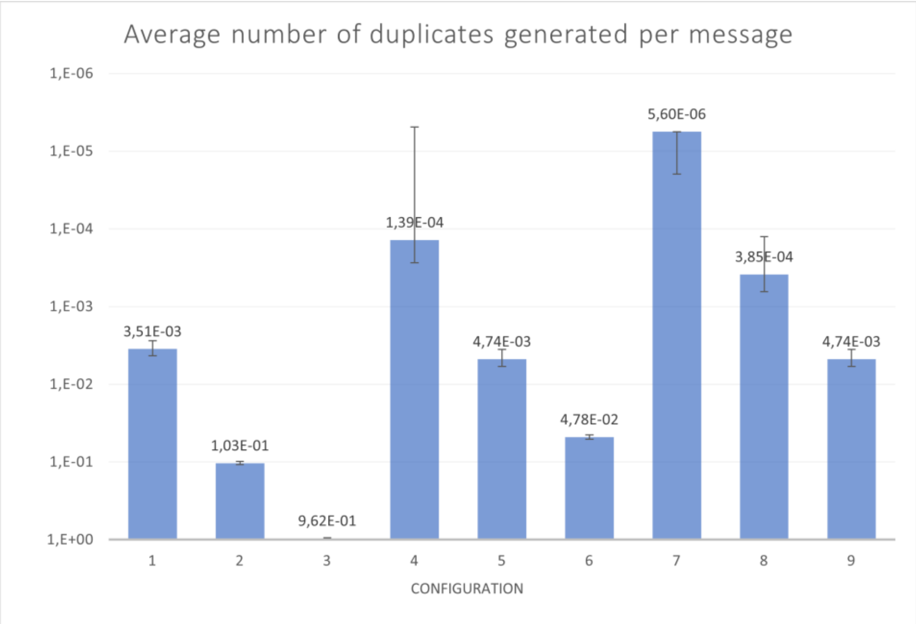


*Figure 9 Duplicates index comparison for different couples of factors D and H.*