Università di Pisa

Computer Engineering

Performance Evaluation of Computer Systems and Networks

Epidemic Broadcast

Group Project Report

TEAM MEMBERS:
Biagio Cornacchia
Gianluca Gemini
Matteo Abaterusso

Academic Year: 2021/2022

Contents

1 Introduction

1.1 Problem Description

From the project group assignment:

*Consider a **2D floorplan** with **N** users **randomly** dropped in it. A random user within the floorplan produces a message, which should ideally reach all the users as soon as possible.*
*Communications are **slotted**, meaning that on each slot a user may or may not relay the message, and a message occupies an entire slot. A **broadcast radius R** is defined, so that every receiver who is within a radius R from the transmitter will receive the message, and no other user will hear it. A user that receives more than one message in the same slot will not be able to decode it (**collision**).*
*Users relay the message they receive once, according to the following policy (**trickle relaying**): after the user successfully receives a message, it waits for a time window of **T slots**. If during that time window it correctly receives less than **m copies** of the same message, it **relays** it, otherwise it **stops**. A sender does not know (or cares about) whether or not its message has been received by its neighbors.*

1.2 Objectives

The aim of the project is to measure and evaluate the **broadcast time** for a message in the entire floorplan, the **percentage of covered users** and the **number of collisions**.

The evaluation is made with the following **parameters**:

- Radius R
- Number of slot T
- Number of copies m
- Number of users N

1.3 Performance Indexes

The main performance indexes are:

- **Broadcast Time**: is the needed time (*seconds*) to reach the maximum coverage in a specific scenario.
- **Number of Collisions**: is how many messages in the same slots occur.
- **Percentage of Covered Users**: let N_c be the number of users that successfully received the message (covered users), the percentage is:

$$\frac{N_c}{N-1}$$

  where $N-1$ is due to the fact that the *message's source* is not taken into account.

Other indexes considered in the analysis are:

- **Number of Connections**: given a Radius, is the total number of links between the users.
- **Percentage of Inactive Users**: let N_i be the number of users that deactivate themselves, the percentage is:

$$\frac{N_i}{N}$$

2 Modeling

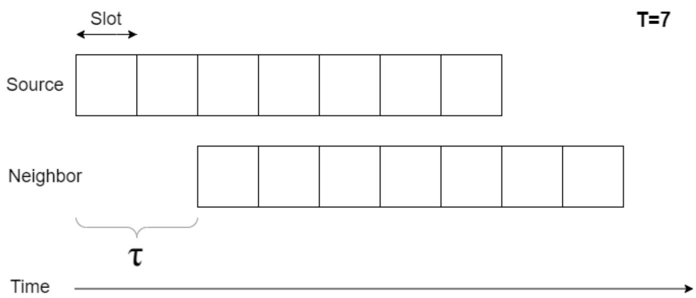
2.1 Introduction

The system is modeled as a 2D floorplan within N users. Each user can communicate with other users if the distance between them is less than a specified radius. In this case they are *neighbors*.
A user (*source*) starts the communication sending the first message to its neighbors. Each of them will relay it to their respective neighbours.

2.2 Assumptions

The following general assumptions have been made:

- The communication is **slotted**, so the slots start at the same time for all users and have the same duration.
- The message transmission lasts **one slot** and starts at the beginning of the slot.
- There are no transmission delay.
- A collision occurs when there are **multiple receptions** in the same slot for a user. If it happens, the collided messages will be **dropped**.
- Each user waits for a **time window** of **T slots** and then checks how many copies of the message have been correctly received (n_c). If $n_c \geq m$ the user **deactivates** itself, otherwise it **relays** the message and waits for the next time window.
- For each user the time window don't start simultaneously, in fact they are misaligned for a factor of τ, in order to avoid that all transmissions fall in the first slot, keeping the slot's beginning synchronized.



- τ and users' positions are uniformly distributed.

3 Implementation

In the implementation the only existing entity is the user.

3.1 User Module

The user module has **bidirectional links** with its neighbors and a couple of coordinates that represents its **position** in the floorplan.
Every user has a own **finite memory** used to store only the messages received in the last time window and their arrival time. It is also used to keep information about the collisions. During the user initialization, the **factor** τ is generated randomly between $[1, T-1]$.

3.2 User Behavior

The user behavior depends on its current state. Three different state are defined:

- **Sleeping**: it's the state before the first successfully received message and it's the default starting state for all users except the source user.
- **Active**: it's the normal operating state for all users. It consists of a check of the message's copies repeated every time window, that allows to decide if the user has to relay the message or deactivates itself.
- **Inactive**: it's the ending state in which the user doesn't relay any message and ignores all the receptions.

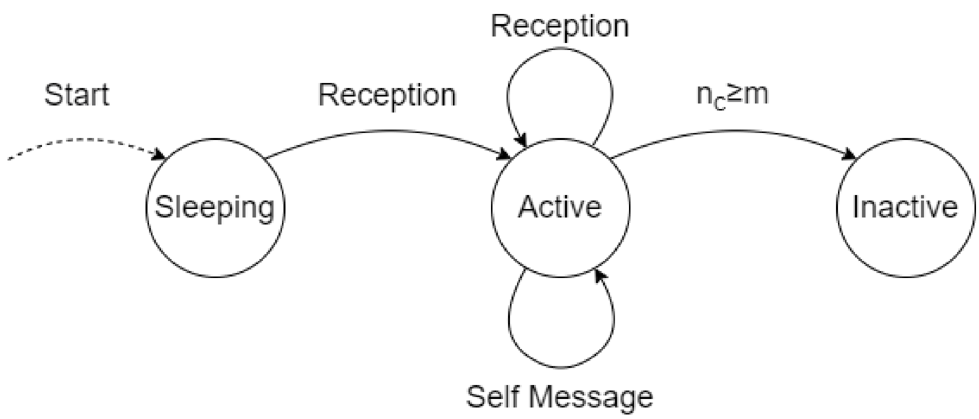


Figure 1: User Finite State Machine

3.2.1 Insight on the Active State

During the active state the possible behavior depends on the type of received message:

- **Self Message**: the user inspects the memory content in order to count the number of successfully received messages. If $n_c < m$ it relays the message to its neighbors and clears the memory, otherwise its state changes to *Inactive*. This message is scheduled every time window until the deactivation.

- **External Message**: each time that a user receives a message by its neighbors, at first it obtains the index of the corresponding slot in the memory using the following statement:

$$Slot_i = \frac{simTime}{t} \bmod T$$

where t is the slot duration. Then, if the slot is empty the user stores the message and its arrival time, otherwise a collision occurs and the message is dropped.

3.3 Dynamic Network

In order to make a completely study of this scenario, it is useful to generate dynamically a network. Two systems have been implemented:

- **Dynamic Radius**: given a specific network consisting of users with a fixed position, the system dynamically generates the links between them, according to the value of the radius.
- **Dynamic Positions**: given the number of the users, the system creates them and assigns positions in line with an uniform distribution. Then it creates the links exploiting the *Dynamic Radius* generation.

The first type of generation is used to test a specific positioning of the users with different radius. The second one is useful to evaluate as many combinations as possible between the number of users and all other possible parameters.

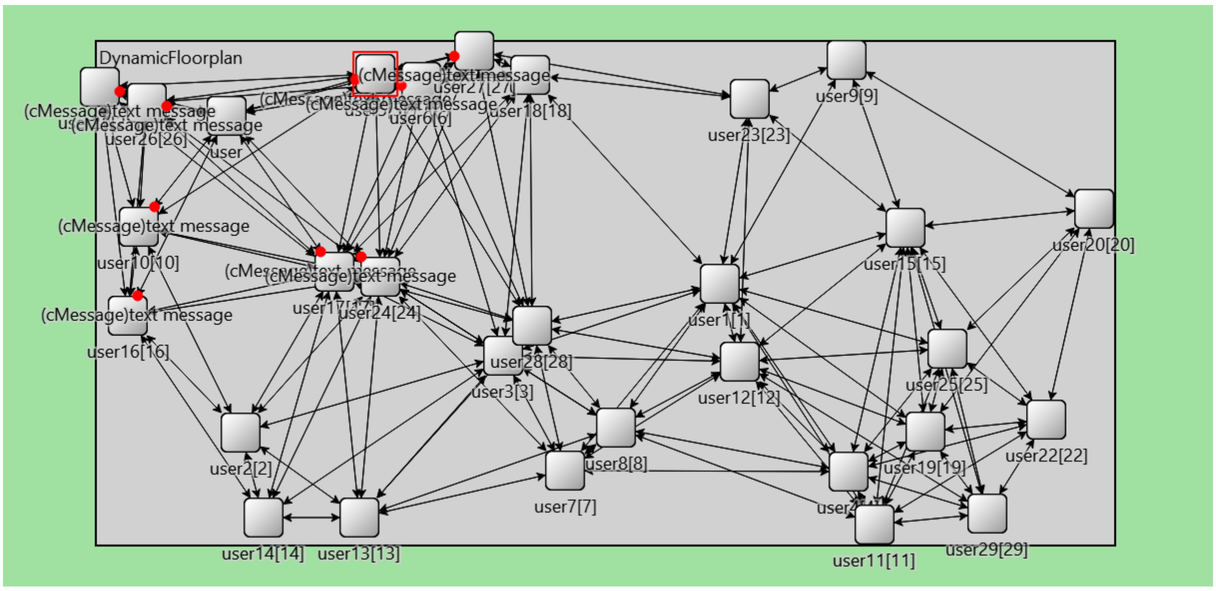


Figure 2: Example of a network dynamically generated with $N = 30$ and $R = 300$

4 Model Verification and Analysis

In this section will be analyzed the effects of the parameters on the network. Below will be introduced some key concepts and terms used in these tests.

- **Unit of measure**: in order to give meaning to the test configurations and results, a value of 100 in Omnet unit corresponds to 1 meter in the real world.
- **Density**: is described by the following formula

$$D = \frac{N}{A}$$

where A is the area of the floorplan $[m^2]$. D is used to generate a realistic network, thus the **maximum chosen value** is one user on square meter.

In the tests will be compared networks with a varying density and floorplan size. In these conditions the parameters values are not comparable, so values that are independent of the above measures are needed. These values are:

- x_R: the values of the radius depend on the diagonal of the floorplan d, therefore this parameter represents the **percentage** of the diagonal.

$$x_R = \frac{R}{d} \qquad x_R \in [0, 1]$$

- x_T: the number of slots should be computed taking into account the user in the network, thus a reasonable choice is to make T dependent on D.

$$x_T = \frac{T}{D} \qquad x_T \geq 0$$

So it is a **multiplication factor** with an inverse unit of measure respect to the density D.

- x_m: the maximum number of copies m can't be greater than the minimum between T and N, and it's expressed as a **percentage** of the minimum between N and the mean number of neighbors of a user.

$$x_m = \frac{m}{min\{N_n, N\}} \qquad N_n = D \cdot R^2\pi \qquad x_m \geq 0$$

At last, a particular case to consider is the **bottleneck** that occurs when an user is the only one that links two different parts of the network. So if this user receives only collisions or gets deactivated, a part of the network will never receive the message.

4.1 Degeneracy Test

The following cases don't allow the correctly working of the system:

- $R = 0$: there are no user with neighbors.
- $T = 0$: it is not possible to receive any message or schedule an event.
- $N = 0$: the network is empty.

- $N = 1$: the network consists only of the source user.

Other particular cases which are kept out the tests are:

- $T = 1$: if an user receives more than one message, it has always collisions, so the only working network is a chain of users.
- $m = 0$: the source transmits once and deactivates itself after T slots, whereas its neighbors deactivates themselves $T + \tau$ slots after the reception. No collisions are possible and the full coverage is reached out only if the source is connected to all the users.
- $m = 1$: it is equal to the previous case but the source never deactivates itself.

4.2 Consistency Test and Insights on Parameters

In this section it is studied how the model works **varying** the parameters and it is checked that the results are **consistent** with the theoretical working of the network. The tests are done **individually** fixing all the parameters except the one under examination.

In the end, for each parameters will be extracted the correct working **intervals** (if it's possible) in which the system shows a stable behavior. During the studies, both dynamic generation typologies are used (Dynamic Positions and Dynamic Radius). Moreover, two different **RNGs** are used to separately control the generation of positions and τ.

4.2.1 Study on R

The radius variation results in the increasing or decreasing of users connections. It is expected that with an R **too small** there will be few connections, thus there will be a lower collisions probability, an higher broadcast time and an higher bottlenecks probability causing a lower percentage of covered users. Instead, with an R **too big** an opposite scenario is expected.

The final goal of this test is to identify the **threshold** over that R is considered too small or too big.

Two configurations are used:

- $N = 12, T = 10, m = 3, R \in [0, 986]$
- $N = 50, T = 35, m = 10, R \in [0, 1118]$

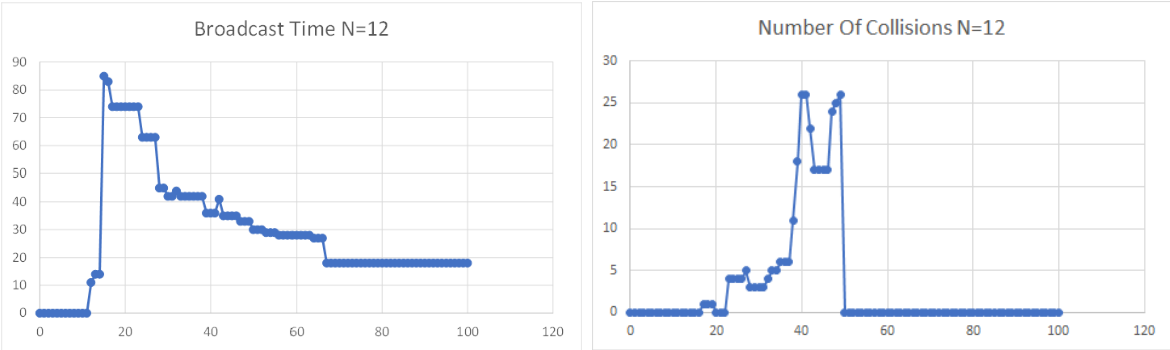
Both configurations use a **fixed seed** for the generation of τ, in order to avoid that the collisions depend too much on randomness.

First Configuration

The first configuration is obtained positioning the users manually and generating the radius R dynamically using the *Dynamic Radius* generation. Being R the parameter in question, T and m are chosen in such a way the **collisions** are not affected too and what it concerns.

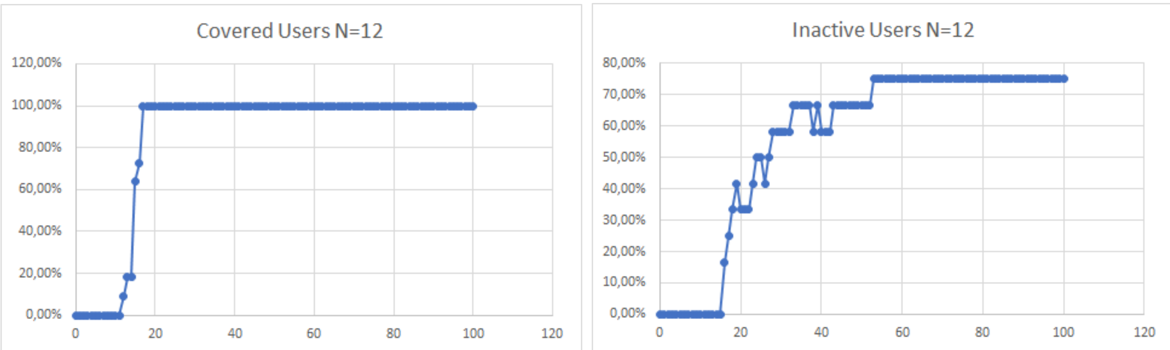
Considering that the floorplan diagonal d is 986 (in Omnet units), the variation of x_R from 0% to 100% causes a variation of the radius R from 0 to 986.

The following charts represent the behavior of the **performance indexes** varying x_R:



(a) (b)

Figure 3: (a) Broadcast Time (b) Collisions



(a) (b)

Figure 4: (a) Covered Users (b) Inactive Users

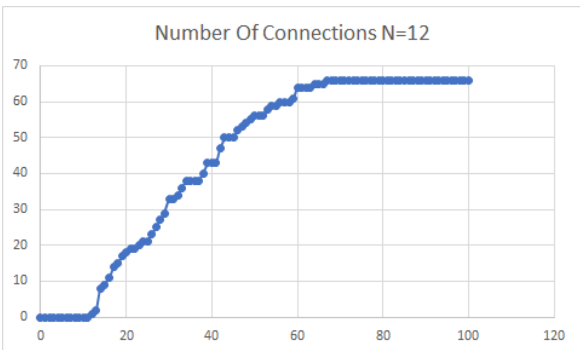


Figure 5: Connections

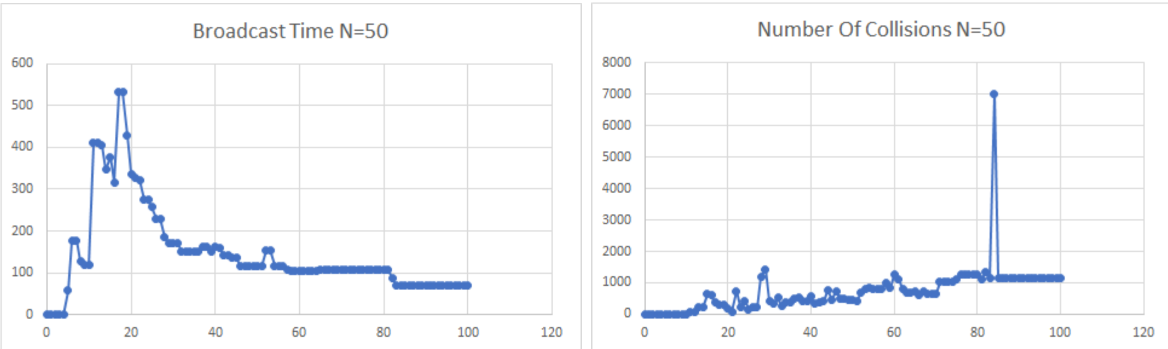
From the charts, it is clear that if the connections grow up the broadcast time decreases and there is an increasing of the coverage, collisions and inactive users.

All the previous hypothesis are met except for the falling of collisions with $x_R > 50\%$ but it's due to an huge number of inactive users (over 70%). Moreover, the rapid increasing of the broadcast time for $x_R < 20\%$ is due to the fact that the maximum possible coverage is less than 100%.

According to the charts, it's evident that with $x_R < 20\%$ the maximum of coverage is not reached whereas $x_R > 70\%$ doesn't bring any benefits (the radius increasing is only a waste of resources).

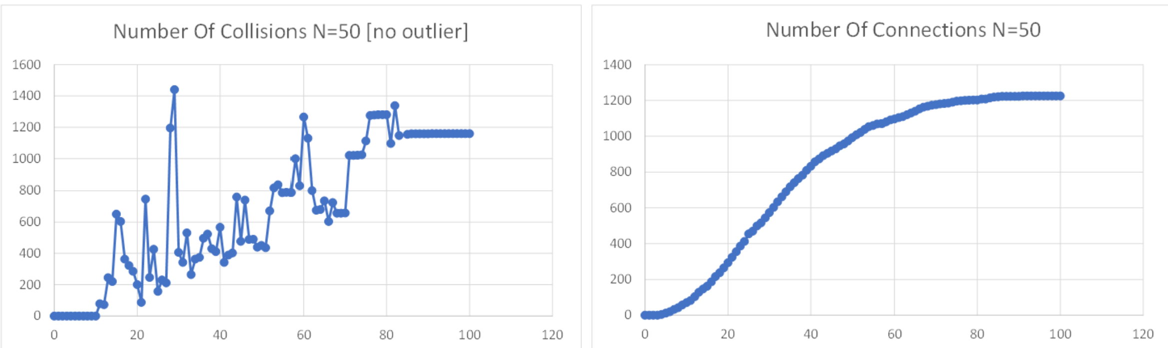
Second Configuration

In this configuration it is used a network dynamically generated using the *Dynamic Positions*, in such a way to see the effects due to the radius increasing in a scenario characterized by the maximum allowed density. The values T and m are chosen in proportion to the values used in the previous configuration. In this case the diagonal d is 1118 (in Omnet units), so the variation of x_R from 0% to 100% causes a variation of the radius R from 0 to 1118.



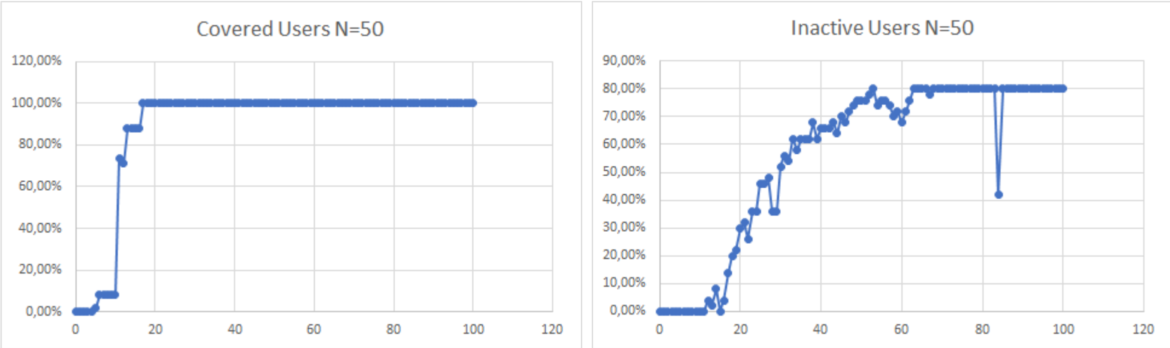
(a) (b)

Figure 6: (a) Broadcast Time (b) Collisions



(a) (b)

Figure 7: (a) Collisions without Outlier (b) Connections



(a) (b)

Figure 8: (a) Covered Users (b) Inactive Users

As in the previous configuration, $x_R < 20\%$ doesn't allow to reach the maximum coverage, whereas for $x_R > 90\%$ there is no benefits. But since near this range the collisions are too high, a good trade-off between the collisions and broadcast time improvements is to choose $x_R < 70\%$. Also in this case the results are consistent with the initial hypothesis.

According to the tests carried out, it is reasonable to consider the following interval for the choice of the radius R:

$$x_R \in [20\%, 70\%]$$

4.2.2 Study on T

The increasing of T results in the increasing of the number of slots between two consecutive transmissions, instead the decreasing of T has the opposite effect. It is expected that with a T **too small** there will be an higher collisions probability because the τ is generated from a smaller interval. Moreover, an higher value of slots causes an higher broadcast time because each user has to wait more time to re-transmit.

Even in this case, the final goal of this test is to identify the **threshold** over that T is considered too small or too big.

Two configurations are used:

- $N = 12, m = 12, R = 425, T \in [2, 28]$
- $N = 50, m = 50, R = 250, T \in [2, 100]$

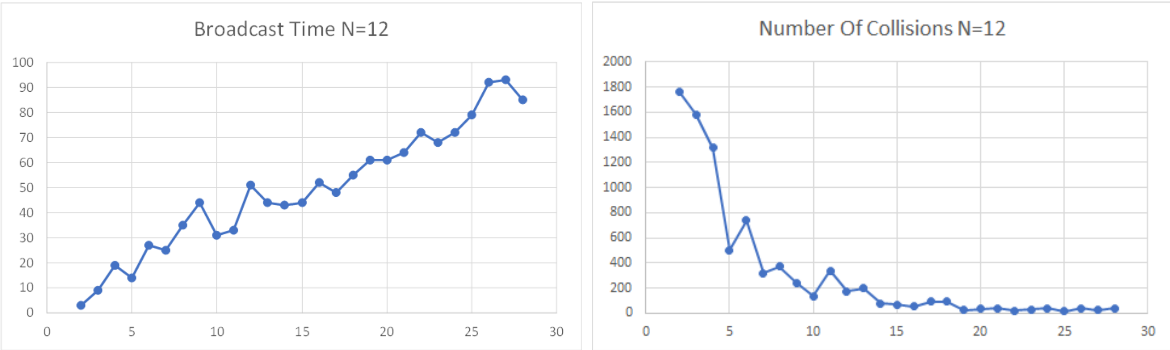
Both configurations use a **variable seed** for the generation of τ, because increasing T at each repetition the generation interval changes.

First Configuration

The first configuration is obtained positioning the users manually and generating the radius R dynamically using the *Dynamic Radius* generation. Being T the parameter in question, R and m are chosen in such a way the **variation of collisions** is mainly dependent on it.

Considering that the floorplan area A is $42,5m^2$ (so $D = 0,28users/m^2$), the variation of x_T from 0 to 100 causes a variation of T from 0 to 28. But according to the degeneracy test, T starts from 2.

The following charts represent the behavior of the **performance indexes** varying T:



(a) (b)

Figure 9: (a) Broadcast Time (b) Collisions

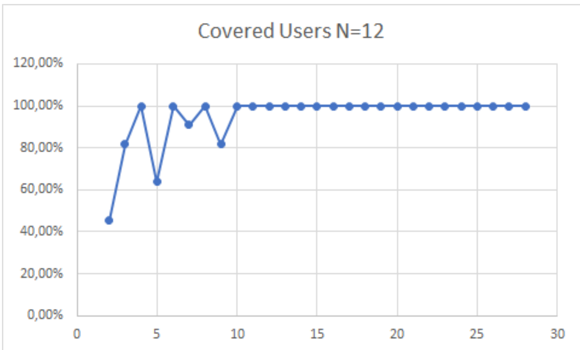


Figure 10: Covered Users

From the charts, it is clear that the broadcast time grows linearly, the coverage reaches the 100% for a $T > 10$ and collisions exponentially decrease. All the previous hypothesis are met.

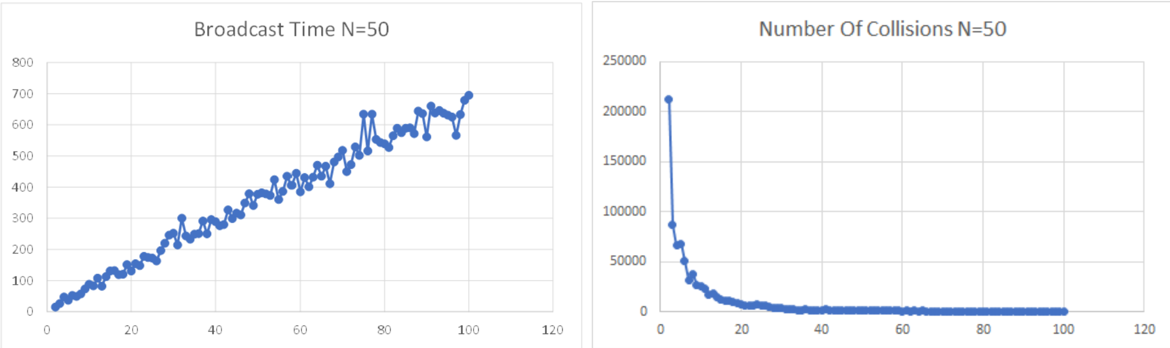
According to the charts, it's evident that with $T < 10$ the maximum of coverage is not reached, whereas $T > 19$ the collisions decrease slowly, respect to the linear and continuous increase of the broadcast time.

So in terms of x_T, the recommended interval for this configuration is $[35, 67]$.

Second Configuration

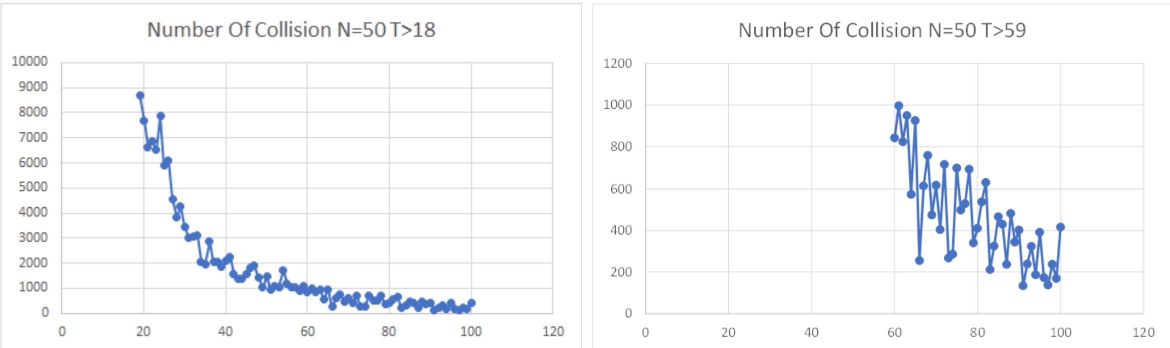
In this configuration it is used a network dynamically generated using the *Dynamic Positions*. As before, the value of m is equal to N. For the radius R, because of the maximum density, it's reasonable to choose an x_R about 20% in order to avoid that the broadcast time is affected too much by the huge number of connections.

In this case the floorplan area A is $50m^2$ (so $D = 1user/m^2$), so the variation of x_T from 0 to 100 causes a variation of T from 0 to 100.



(a) (b)

Figure 11: (a) Broadcast Time (a) Collisions



(a) (b)

Figure 12: (a) Collisions for $T > 18$ (a) Collisions for $T > 59$

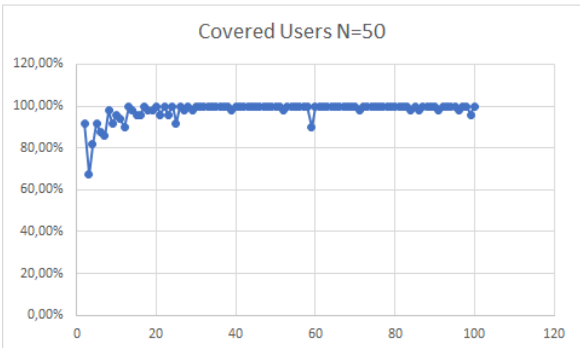


Figure 13: Covered Users

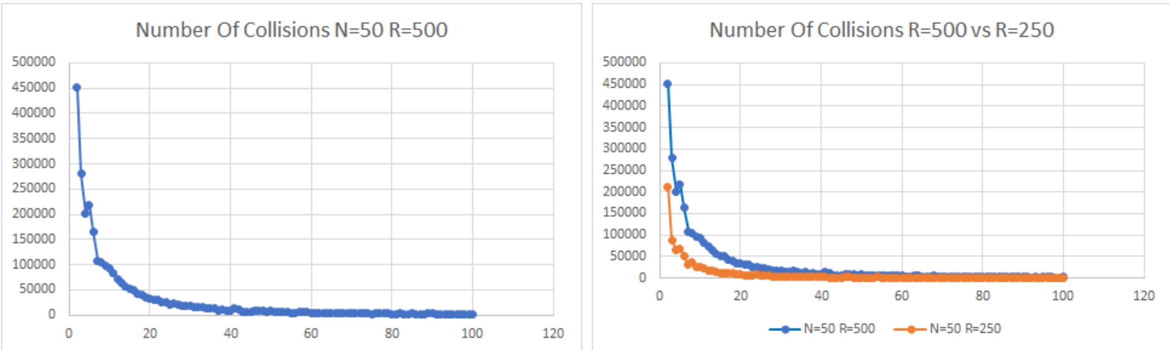
From the charts $T < 20$ doesn't allow to reach the maximum coverage but, by enlarging the chart in Figure 11, it is possible to see that the collisions are too high, then a good choice can be an $T > 40$. Instead, for $T > 80$ there are no considerable improvements considering that the broadcast time linearly grows up. So the suggested interval for this configuration is $x_T \in [40, 80]$.

Also in this case the results are consistent with the initial hypothesis.

Comparing the results obtained from the two configurations, it is noticed that the collisions trends are the same. But it's clear that the collisions charts (and so the intervals of x_T) appear translated and scaled in Figure 9 and 11. So it's necessary to figure out why this phenomena happens.

Other Configurations

In order to understand what causes the previous phenomena, the second configuration has been modified doubling the radius ($R = 500$).

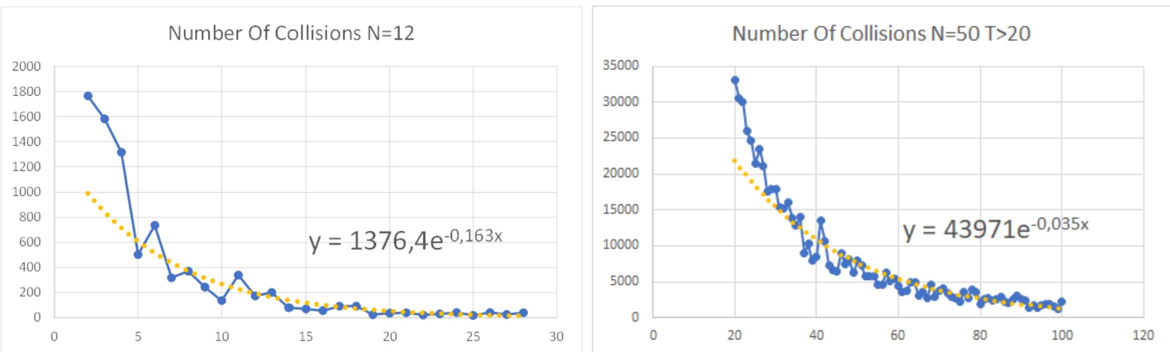


(a) (b)

Figure 14: (a) Collisions for $R = 500$ (b) Radius Comparison

In this case, it's possible to see a translation to the right. A plausible hypothesis is that the translation is mainly influenced by the radius while the scale is mainly influenced by the number of the users.

To verify this, in the previous configuration the number of users has been increased to 70 and compared with the others results.



(a) (b)

Figure 15: (a) Collisions for $N = 12$ (b) Collisions for $N = 50$

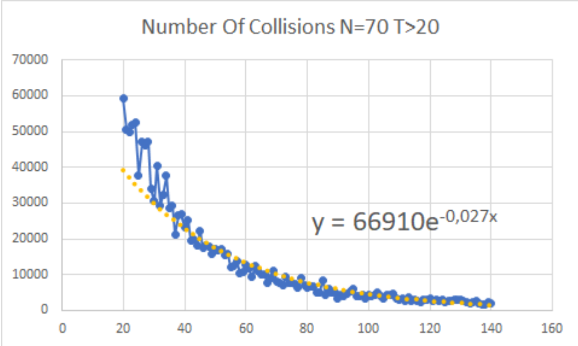


Figure 16: Collisions for $N = 70$

In conclusion, increasing N, the velocity at which collisions tend to 0 decreases. So the hypothesis are verified.

According to the tests carried out and considering the dependency on N and R, it is not possible to suggest a priori interval of x_T.

4.2.3 Study on m

The increasing of m means that a larger number of the copies to deactivate the user is needed. So, if m is **too big** the users never deactivate themselves and the probability of reaching the maximum percentage of coverage is higher. But this involves a waste of resources because of the unnecessary re-transmissions. Moreover, an high value of m increases the probability of collisions. On the other hand, an m **too small** may cause an higher probability of generating bottlenecks.

Even in this case, the final goal of this test is to identify the **threshold** over that m is considered too small or too big.

Two configurations are used:

- $N = 30, R = 300, T = 30, m \in [2, T]$
- $N = 50, R = 200, T = 50, m \in [2, T]$

Both configurations use a **fixed seed** for the generation of τ, in order to observe the influence that m has on collisions, reducing as much as possible the randomness. For the same reason, it's chosen $T = N$.

Instead, each configuration has a **different seed** for the generation of the positions (that remains **fixed** for every repetition).

The choice of R has been made based on the density D of the users, in such a way to not having a level of connection that might hide the effects of m in the network.

The variation of m between 0 and T has been studied for each configuration.

For the first configuration, N it's chosen equal to 30 because for small N the parameter m has no meaningful effect on the system. In fact, the Figure 17 shows that the system is unstable if the parameters T and R are less influential (condition needed for the study of m).

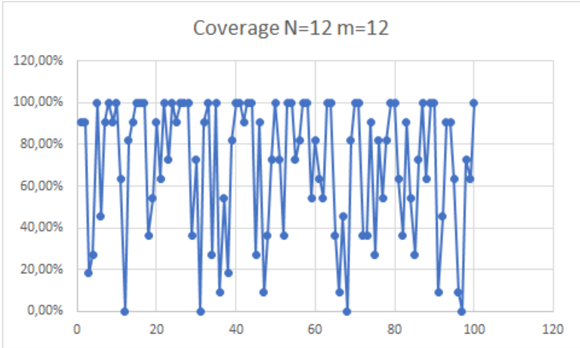
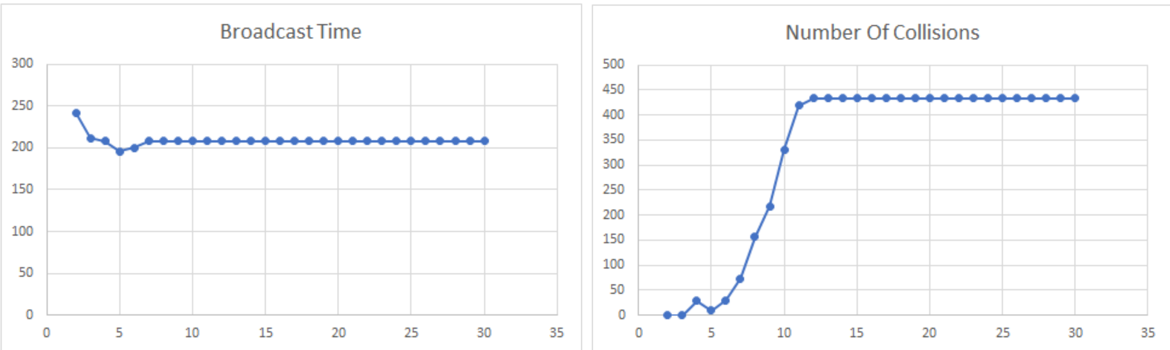


Figure 17: Coverage of the same network in 100 repetitions with different τ

First Configuration

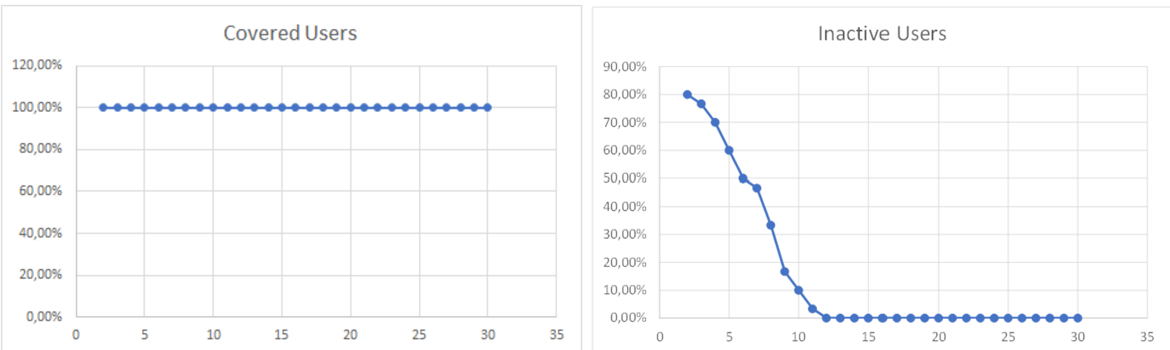
For the first configuration it is used a network dynamically generated using the *Dynamic Positions*.

The following charts represent the behavior of the **performance indexes** varying m:



(a) (b)

Figure 18: (a) Broadcast time (b) Collisions



(a) (b)

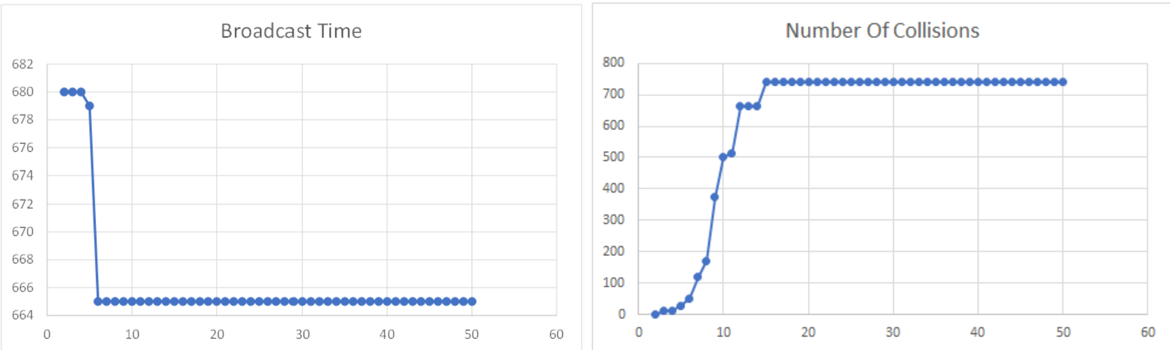
Figure 19: (a) Covered Users (b) Inactive Users

From the charts it's clear that the total coverage is always reached, but for $m < 4$ the broadcast time increases. Instead, for $m > 12$ there are not changes in the number of deactivated users. So, the suggested interval for x_m is $[23\%, 71\%]$. All the previous hypothesis are met.

Second Configuration

Even in this configuration it is used a network dynamically generated using the *Dynamic Positions*.

The following charts represent the behavior of the **performance indexes** varying m:



(a) (b)

Figure 20: (a) Broadcast time (b) Collisions

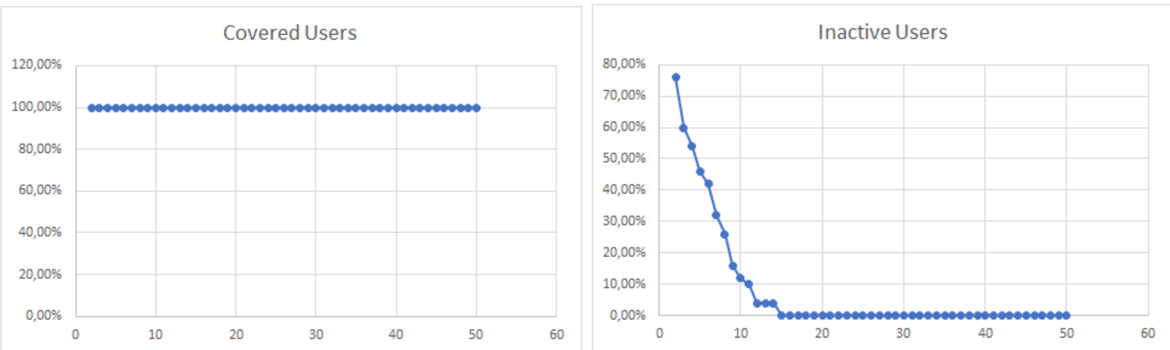


Figure 21: (a) Covered Users (b) Inactive Users

The charts are consistent with the previous configuration. For $m < 6$ the broadcast time is higher while for $m > 15$ any user deactivates itself. So, the suggested interval for x_m is $[46\%, 115\%]$.

It's evident that for the first configuration a $m = 12$ is enough to avoid the deactivation for every user, while for the second one $m = 15$ is needed. As expected, the required value of m is increased because the mean number of neighbors for user is grown up.

Moreover, it's impossible to identify a priori an interval for x_m, since it strongly depends on the chosen radius.

4.2.4 Study on N

The last remaining parameter to analyze is N. It has already the maximum value ($D = 1user/m^2$). Now it's necessary to find its minimum value. For this goal, the following configuration has been chosen:

- $R = 300, T = \frac{N_{max}}{2}, m = T * 0.25, N \in [2, 50]$

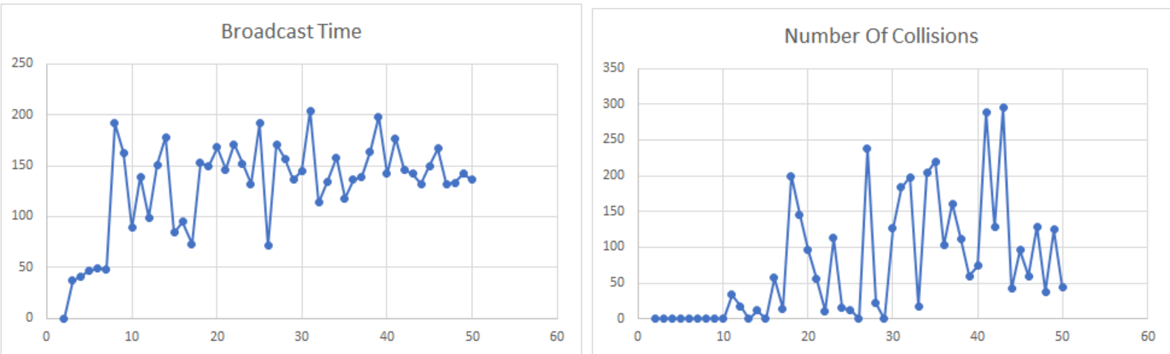
The radius R should not exceed the 50% of the diagonal d because it's needed to see the effects of an N too small, otherwise the maximum coverage will be always reached.

Moreover, T has been chosen in such a way that it's possible to see the effects of N on the collisions (varying N there are either values of $N < T$ and $N > T$).

In the end, m has been chosen in such a way that it's possible to see the effects of N on the users deactivation.

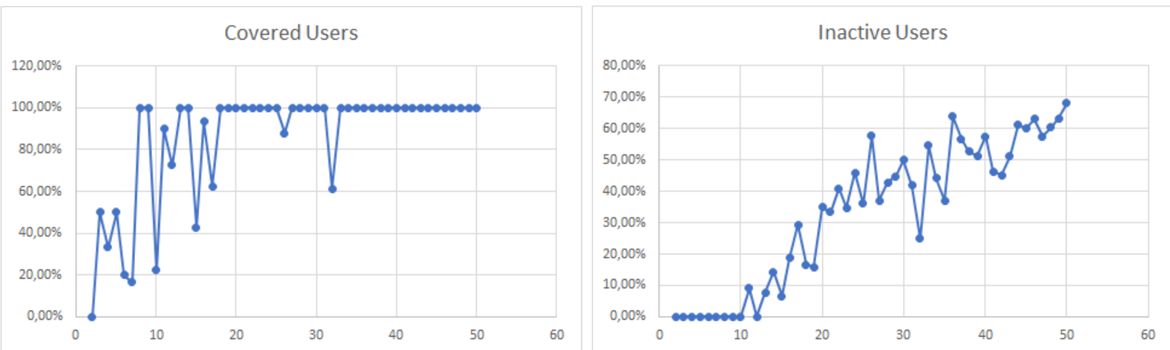
It is expected that increasing N, the connections grow up and consequently the percentage of coverage, the collisions and users deactivation will increase in the same way.

The charts obtained are the following:



(a) (b)

Figure 22: (a) Broadcast time (b) Collisions



(a) (b)

Figure 23: (a) Covered Users (b) Inactive Users

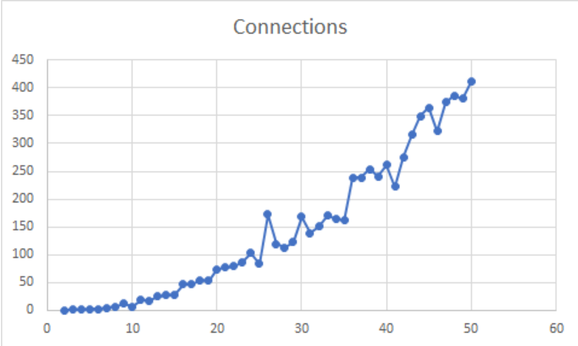


Figure 24: Connections

As expected, since the connections keep growing with N, it's possible to see how the percentage of coverage gradually increases until it becomes stable (after about $N = 20$). Therefore, the number of deactivated users increases, but the collisions don't. The latter is due to the trade-off between:

- the growing number of connections that increases the probability of collisions
- the growing number of deactivated users that decreases the probability of collisions

To verify this, the same simulation has been carried out, but with $m = 50$ (the maximum possible value for N), to remove the effect of the deactivated users:

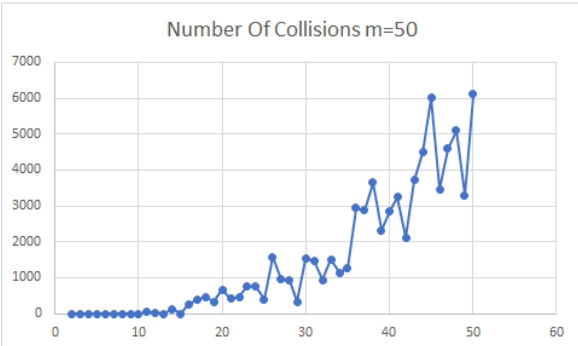


Figure 25: Collisions with $m = 50$

As expected, without the effect of deactivated users, the collisions keep growing with N.

Looking at the charts in Figure 22 and 23, it's clear that for an $N < 10$ there is a low coverage, while for $N > 20$ the coverage is stable. So, in terms of density D it is strongly discouraged to use $D < \frac{1}{5}$. Instead, to achieve a stable coverage is suggested to choose $D > \frac{2}{5}$.

So, recalling what said in the introduction of Chapter 4 about the maximum density possible, the suggested interval for D is $[\frac{2}{5}, 1]$.

5 $2^k r$ Factorial Analysis

After analyzing the parameters individually, their combined effect on the **KPI** has been studied.

For each configuration 100 simulations have been made, in order to obtain a meaningful data set.

All the four parameters (R, T, m, N) have been analyzed using 3 repetitions. The results have been evaluated with a CI of 95%.

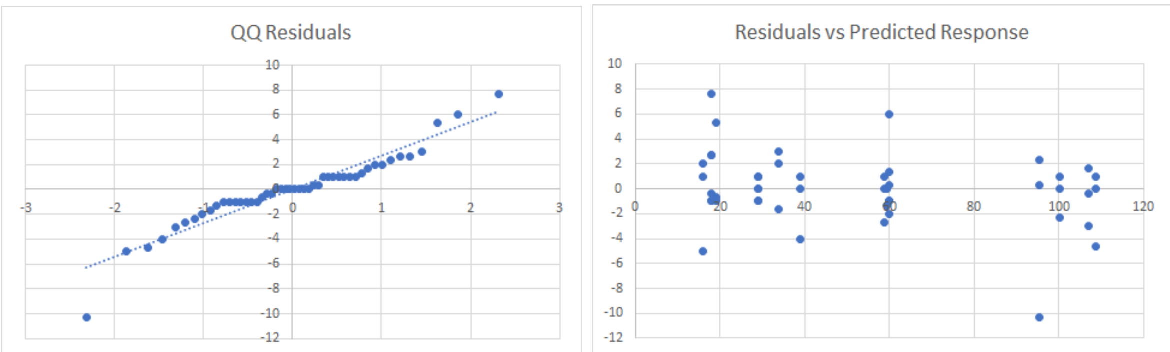
5.1 Broadcast Time Analysis

The parameters have been chosen in order to verify the broadcast time variation, in such a way that all of them can influence the network.

In particular, the configurations consist of a low and medium density network (however, the low density is greater than $\frac{1}{5}$). The radius belongs to the interval computed in the paragraph *4.2.1*, in such a way the behaviour of the broadcast time can be varied, keeping it stable.

- R = $\{300, 650\}$
- T = $\{6, 20\}$
- m = $\{3, 6\}$
- N = $\{12, 30\}$

Firstly, it is necessary to verify that the results are meaningful, then residuals have to follow a normal distribution and their standard deviation have to be constant.



(a) (b)

Figure 26: Plots for testing the hypothesis

The first graph (Figure 26) shows a linear tendency, that confirms the normal distribution hypothesis. Instead, the second one shows no visible trend, then even the standard deviation hypothesis is verified.

The results are the following:

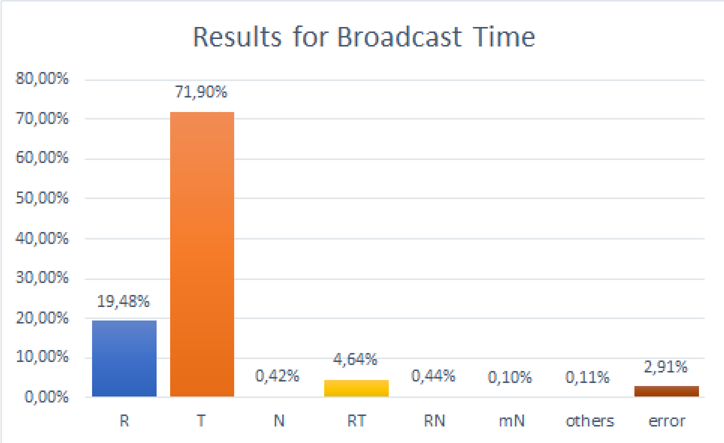


Figure 27: $2^k r$ Results

From the histogram (Figure 27), it's clear that the broadcast time mainly depends on the parameter T. This is obvious since increasing T, the time between the retransmissions grows up. This implies that the necessary time to get to all the reachable users increases as well.

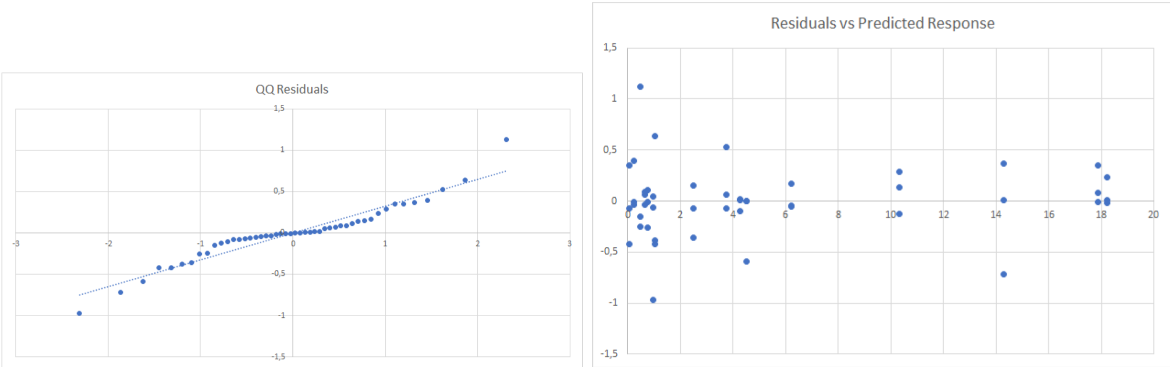
Secondly, even R has a discrete influence since the higher is the number of connections, the lower is the time needed to reach all users.

5.2 Number of Collisions Analysis

For this analysis, the parameters are the same of the previous one since they allow to correctly represent the variation of collisions varying all four (R, T, m, N).

- R $= \{300, 650\}$
- T $= \{6, 20\}$
- m $= \{3, 6\}$
- N $= \{12, 30\}$

It's necessary to verify that the results are meaningful. The values recorded for the $2^k r$ factorial have different scales of magnitudes, especially comparing extremes with density and radius quite different. So, in order to obtain a normal distribution of the residuals, the transformation $y = log^2 x$ has been applied, yielding the following results:



(a) (b)

Figure 28: Plots for testing the hypothesis

The charts show that both hypothesis are met. The results are the following:

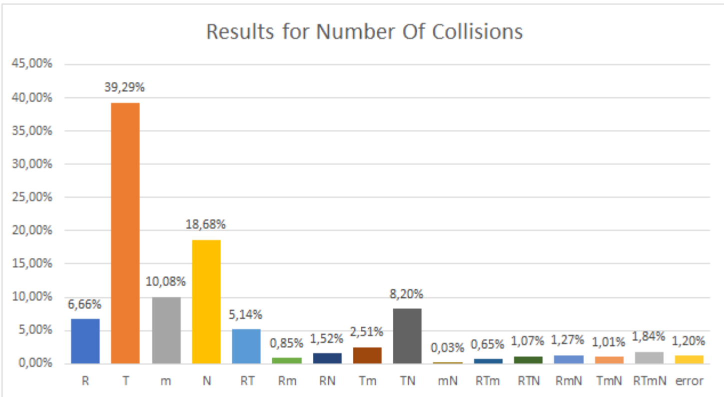


Figure 29: $2^k r$ Results

The Figure 29 shows that the most influential parameter is T, in fact increasing the number of slots, the range of values that τ can assume becomes larger and then the probability of generating the same τ decreases.

Another parameter to take into account is N. A larger value of N implies an higher density D and so an higher number of connections. This means that an users will potentially receive more messages and consequently the probability of collisions will be higher.

Even a combination of these two parameters has a strong influence since an higher density D requires an higher value of T (as shown before), keeping R fixed.

Then, another important parameter is m which, assuming low values, causes an high number of deactivated users. This implies a lower number of users that can cause collisions (during the rest of the simulation).

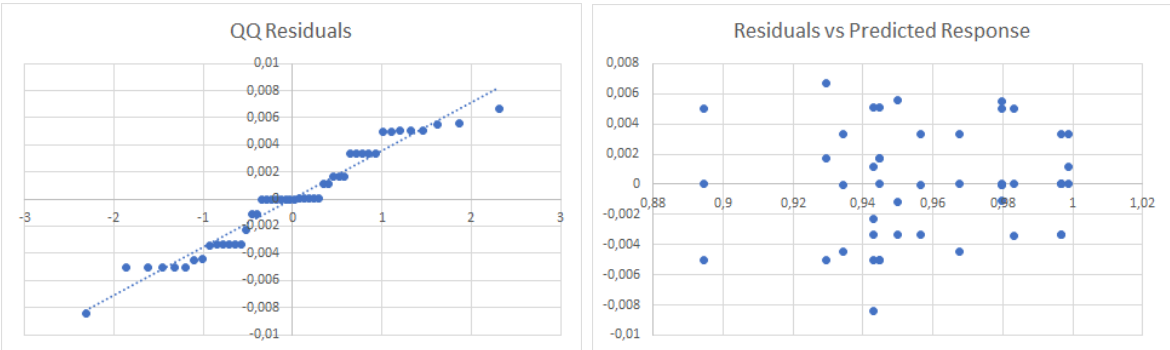
Finally, the last parameter is R. If it increases, the number of connections grows up with less impact respect to N.

5.3 Covered Users Analysis

The study on the percentage of covered users shows a problem. In fact, in a network of 12 users the possible percentages are 12 ($\frac{0}{12}$, $\frac{1}{12}$, ...). Through these few values the model is weak and the hypothesis on the IID normal residuals and standard deviation are not met. So, in order to obtain an higher variety between the recorded values, it's needed to scale the network using huge N and so an higher area for the floorplan.

- R = $\{650, 800\}$
- T = $\{10, 20\}$
- m = $\{4, 9\}$
- N = $\{200, 300\}$

To verify that the results are meaningful, the analysis are shown below:



(a) (b)

Figure 30: Plots for testing the hypothesis

The charts show that both hypothesis are met. The results are the following:

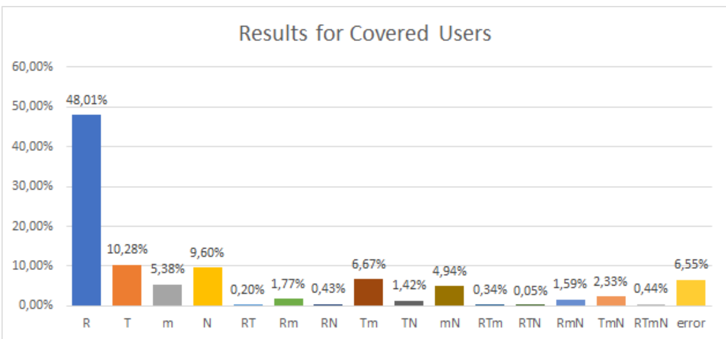


Figure 31: $2^k r$ Results

From the histogram, it's evident that the most influential parameter is the radius R. In fact, a big enough radius allows to reach all the users in the first transmission. Generally, if the radius is high, the number of users to cross to get the broadcast will be lower, so the probability of failing a transmission is lower.

6 Conclusions

From the results obtained in the measurements can be stated that the choice of the parameters, which influences the broadcast working, has to be weighted respect to the requirements wanted. In particular, the main trade-off is between the number of collisions and the broadcast time:

- If a **low broadcast time** is required, T should not be too high, in order to reduce the delay between the retransmissions. An effect of this choice can be the increasing of the probability of collisions.
- If a **low number of collisions** is required, should be necessary to increase T, and consequently the broadcast time will increase as well.

In general, to have a stable network it's necessary an $x_R \in [20\%, 70\%]$ and a $D \in [\frac{2}{5}, 1]$, but in some cases the parameter N is given and not manageable, so the latter interval may not be met.

- If a $D < \frac{2}{5}$ is required, the network can be stabilized using a big enough radius, keeping in mind the consequent increasing of the number of collisions caused by the growth of connections.
- If a $D > 1$ is required, a small radius should be used, in order to decrease the number of collisions, taking into account the likely decreasing of coverage and the increasing of the broadcast time.

In the last two cases, it's clear the dependency between connections and collisions. In fact, the increasing of connections to reach a **better coverage** could yield the opposite effect, due to the higher value of collisions. In this scenario the most significant parameter is T, which allows to keep down the number of collisions and to increase the number of deactivated users.

In the end, if the epidemic broadcast has to be applied to a **low consumption network**, it is important considering the parameter m. A too low m can early block the network, due to the massive deactivation of the users. Instead, if it's too high can cause a low number of deactivations, so a waste of resources.