Università di Pisa

Performance Evaluation of Computer Systems and Networks - Project: Epidemic Broadcast

Fabio Piras, Giacomo Volpi, Guillaume Quint

Academic Year: 2022/2023

Contents

1 Introduction **3**
1 .1 Problem Description 3
1 .2 Performance Indexes 3
1 .3 Parameters and assumptions 3

2 Implementation **4**
2 .1 Node Behaviour 4
2 .2 StatCollector Behaviour 5

3 Case Study: 200 Nodes **6**
3 .1 Introduction 6
3 .2 Performance indexes 6
3 .3 Hierarchy of priorities 10
3 .4 Time evolution 10
3 .5 Factorial Analysis of ending time and ending coverage 11

4 Model fitting **14**

5 Testing the assumption for the Gompertz model **15**

6 Varying node number **17**
6 .1 50 and 700 cases 17

7 Factorial analysis on different number of nodes **22**
7 .1 Influence on coverage rate and time 22

8 Conclusions **25**

A Appendix **26**
A.1 50 Nodes 26
A.2 100 Nodes 27
A.3 200 Nodes 28
A.4 500 Nodes 29
A.5 700 Nodes 30

1 Introduction

1 .1 Problem Description

N nodes are randomly placed on a 2D *floorplan*. A random node within the floorplan produces a *message* that should reach all users as soon as possible. Communications are slotted. Each node has a broadcast radius R defined so that only the receivers within that radius from the transmitter receive the message. A node that receives more than one message in the same slot experiences a collision and it is not able to decode any of them.

After a node successfully receives a message, it keeps extracting a value from a Bernoullian RV with success probability p on every slot, until it achieves success, after which it relays the message and stops.

1 .2 Performance Indexes

The performance of a given configuration is defined through different performance indexes, namely:

- the time it takes a message to reach all nodes, or alternatively, the time after which every node is either stopped or waiting for a message
- the percentage of nodes that have received the message without collisions
- the mean percentage of nodes that experiences a collision in a given slot

1 .3 Parameters and assumptions

The floorplan is assumed to be square and with a fixed dimension of side $L = 100$ units.

Nodes are initialized with coordinates uniformly distributed on the x and y axis ranging from 0 to L.

For simplicity, each time slot lasts exactly 1 second. Communication between nodes is idealized and because every message must be transmitted before the next time-slot, each transmission is assumed to last exactly 250 ms, regardless of the actual distance between nodes.

Different simulation campaigns have the following varying parameters:

- N number of nodes, which is an indicator of population density for a given floorplan area $[50, 100, 200, 500, 700]$
- R units of radius for each and every node. This is a constant for each node in the same simulation $[10, 30, 50, 75, 100]$ (Note that a radius of 100 still isn't a limit case where every node is certainly connected)
- p probability of success at each time-slot for a node ready to transmit a message

  $[15\%, 30\%, 50\%, 70\%, 85\%, 100\%]$

For each combination of these parameters, we performed 33 repetitions, in order to have statistically sound scenarios from which we could obtain reasonably meaningful and valid data.

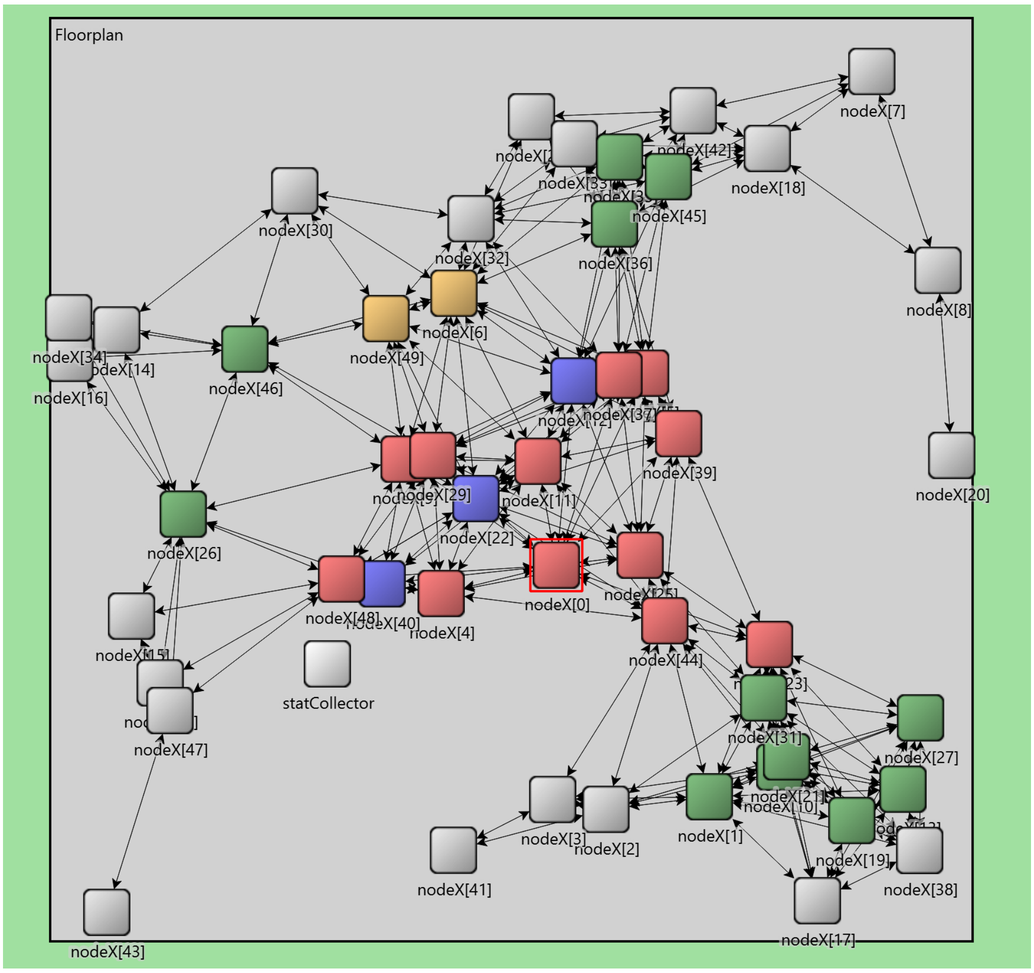


Figure 1: debugging view of the simulated floorplan

2 Implementation

The following modules have been implemented:

- `Node` that can receive and transmit messages according to the specifications
- `StatCollector` that retrieves statistical information from the system at every time slot.

For debugging purpose, a color code inside omnetpp's graphical interface has been chosen to reflect the status of each node at every instant, as can be seen in Figure 1

2 .1 Node Behaviour

A node can be found on 5 different statuses

- *Waiting.* (gray) Node is listening for an upcoming message from one of its neighbors.
- *OneMessage.* (blue) Node received exactly one message in this time slot. At the end of it, if it doesn't receive any other message, it will perform the random experiment as described in the Ready state, otherwise it will be a Collision.
- *Collision.* (orange) Node received more that one message in this time slot. In the next time slot it will restart its cycle from the Waiting state.

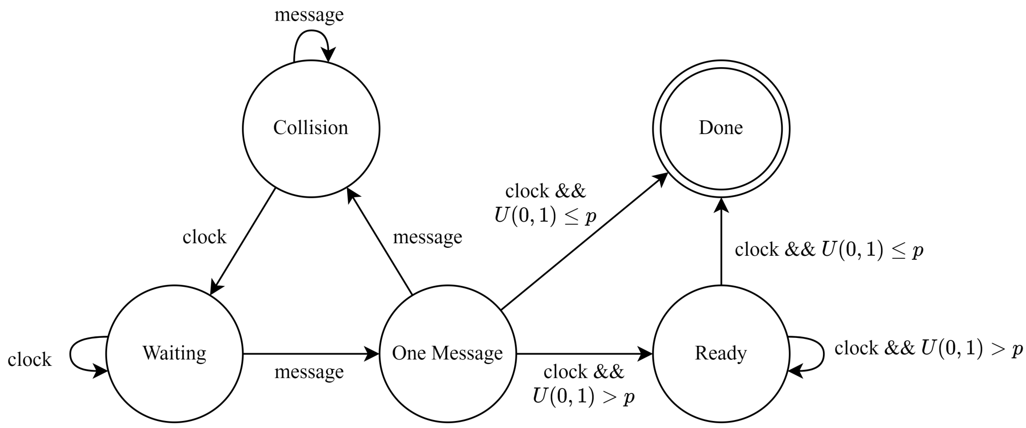


Figure 2: Finite State Machine Diagram for the Node component

- *Ready.* (green) Node has a valid copy of the message. At the start of every time slot the Node extracts a uniformly distributed RV $\sim U(0,1)$ and confronts it with a constant threshold p. If it succeeds, it relays the message to its neighbors and transitions to the Done state, regardless of it having generated a collision or not, otherwise it remains in the Ready status. Over multiple time-slots, this implements the event of status transitions as a Bernoullian distributed RV with probability p.

- *Done.* (red) The node has completed its cycle. It won't respond to any more incoming messages.

Synchronization between nodes is idealized and isn't modeled. Each node starts its communication exactly at the beginning of the current time-slot.

2 .2 StatCollector Behaviour

This module is a singleton for each simulation and isn't accessible to the rest of nodes. Because we assumed that every communication completes after 250 ms (the delay of transmission), this modules executes with the same slotted frequency as all other nodes but with an offset of 500 ms. This guarantees that the information retrieved by `StatCollector` comes after every node has already transitioned to the new status.

This module saves the relative distribution of nodes between the different statuses, i.e. what fraction of the system is in what state for each time-slot.

3 Case Study: 200 Nodes

3 .1 Introduction

In this chapter we analyze the case at 200 nodes, as it is the quantity that is halfway between our configurations. At the end of the simulations we decide to calculate the average coverage percentage, summarized in this table.

	r=10	**r=30**	**r=50**	**r=75**	**r=100**
p=0.15	0,9073	1	1	1	1
p=0.3	0,8938	0,9992	0,9956	1	0,9992
p=0.5	0,7991	0,9942	0,9880	0,9889	0,9959
p=0.7	0,7105	0,9798	0,9645	0,9373	0,9903
p=0.85	0,5730	0,9421	0,9021	0,9183	0,9797
p=1	0,4755	0,8191	0,6712	0,7630	0,9673

Table 1: Mean coverage percentage

The following table show the mean coverage time for each run.

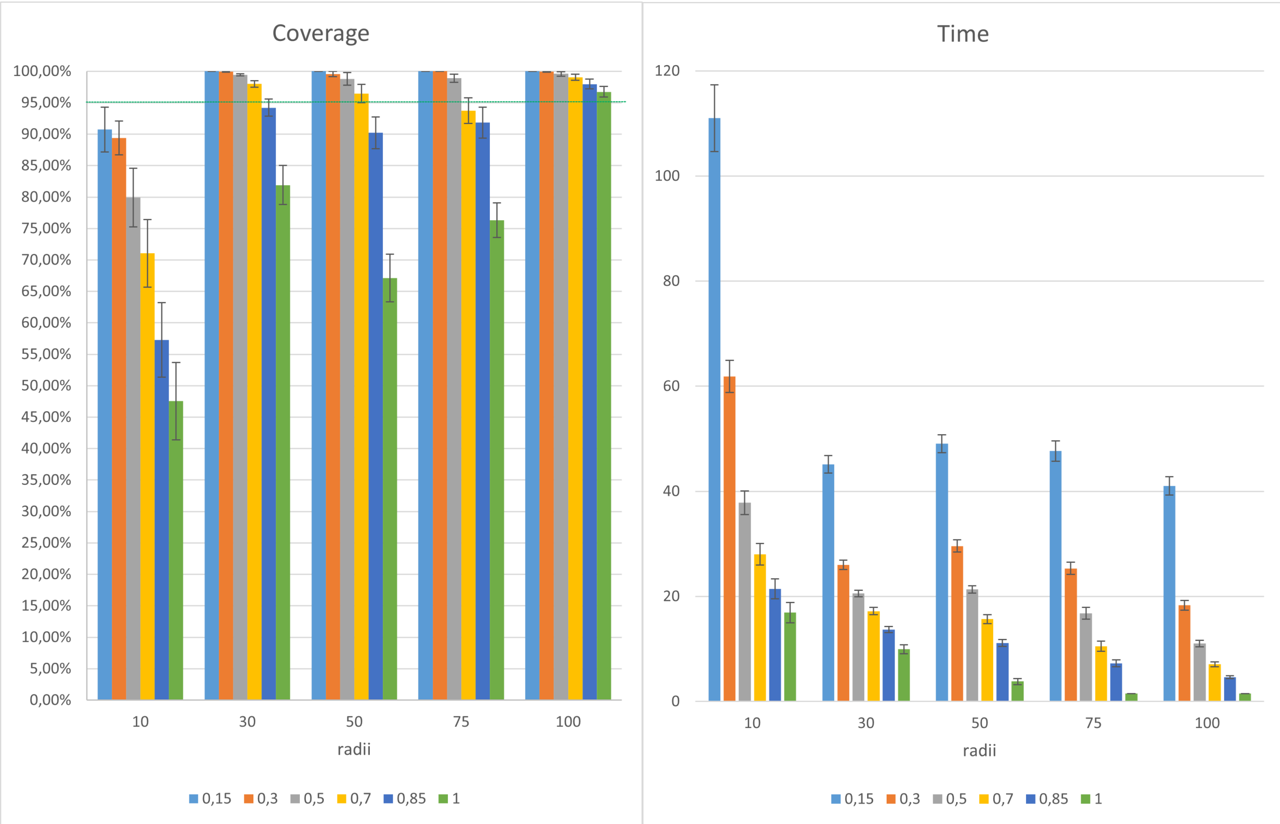
	r=10	**r=30**	**r=50**	**r=75**	**r=100**
p=0.15	111,0152	45,1364	49,0455	47,6515	41,0152
p=0.3	61,8636	26,0152	29,5909	25,3182	18,2879
p=0.5	37,8333	20,5606	21,3182	16,7727	11,0455
p=0.7	28,0152	17,1970	15,6515	10,5303	7,0758
p=0.85	21,4091	13,6818	11,1061	7,2576	4,6515
p=1	16,8939	9,9242	3,8030	1,5	1,5

Table 2: Mean coverage time

Since the means do not include any information on the variance of the data, we decide to plot a column graph showing the 90% confidence interval for the average coverage values (Fig. 3). Note that we chose to discard all runs that do not end with a coverage percentage above 95% with a 90% coverage. That's why in the following boxplot the runs with radius 10 are not shown.

3 .2 Performance indexes

After selecting the runs to be analysed, we choose to plot the most interesting performance indicators such as: the coverage percentage (Figure 4), the completion time (Figure 5) and the mean percentage of collisions for each time-slot (Figure 6). For symmetry, the graphs also show configurations, such as the one with probability 0.7 and radius 75, even if they should not be considered according to the previously set standards.



(a) Coverage percentage (b) Coverage time

Figure 3: Column graph with 90% CI

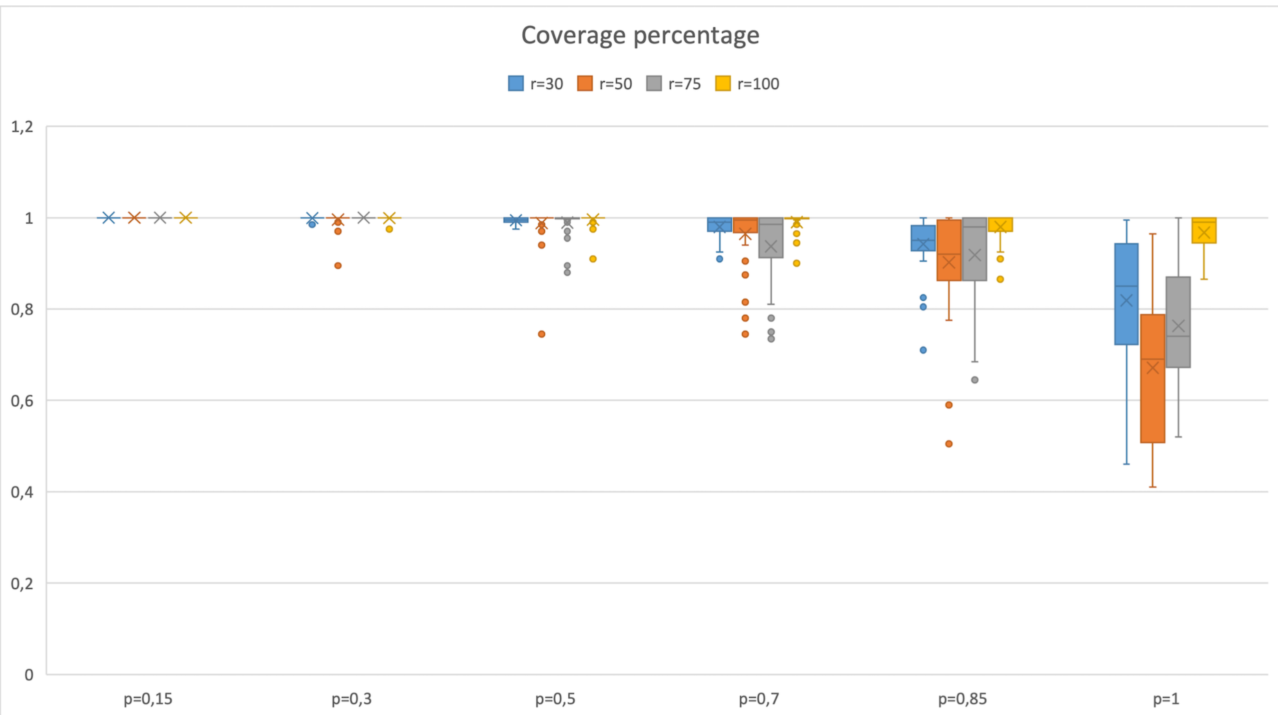


Figure 4: Coverage percentage

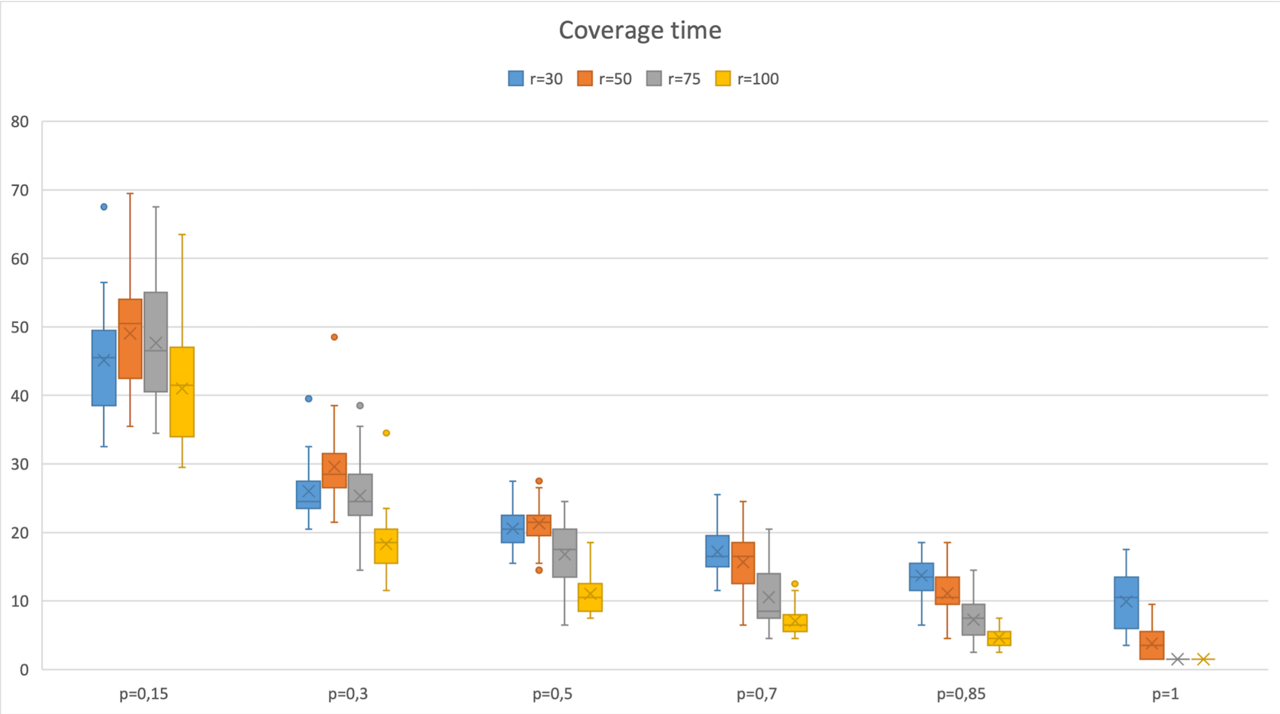


Figure 5: Coverage time

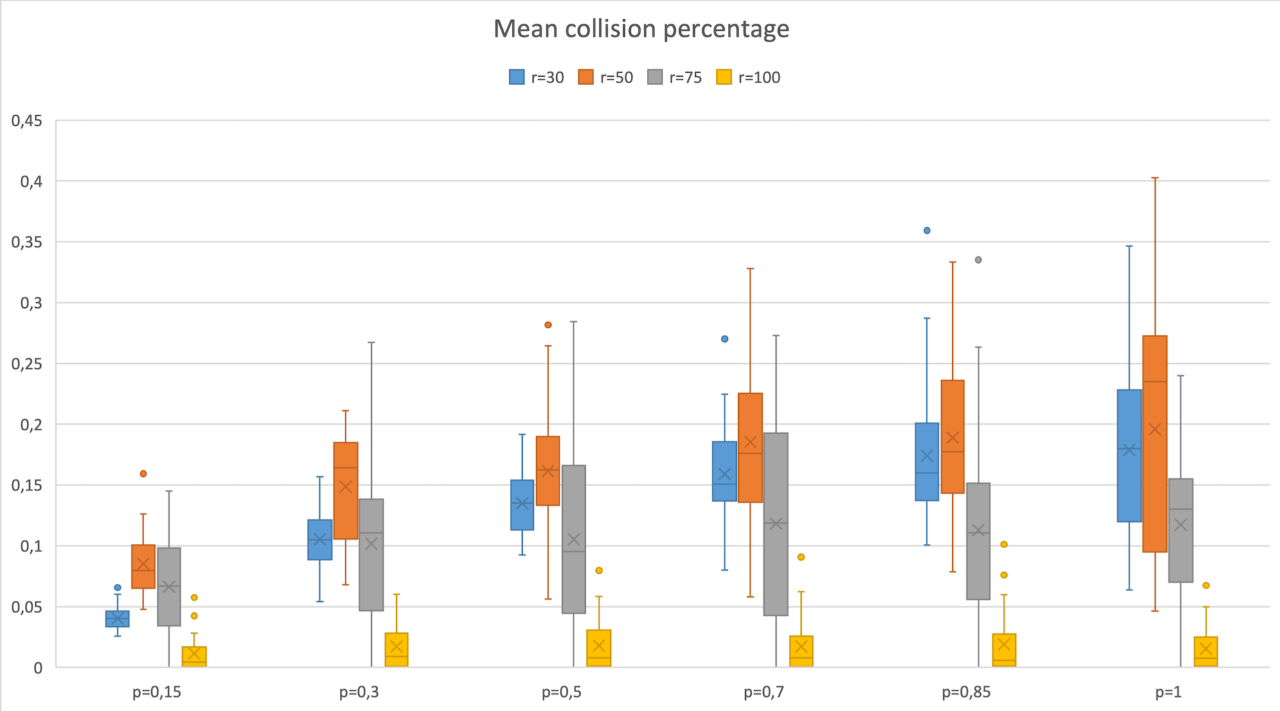


Figure 6: Mean collision percentage

Looking at coverage percentage graph (Fig. 4), it can be seen that increasing the probability does not give better coverage in most cases, as one would expect. In fact, it should be emphasized that as the probability increases, collisions also tend to increase accordingly. As imagined, the coverage time (Fig. 5) decreases as the probability increases, as a higher probability translates into more nodes that can send the infection message. Regarding the collision index (Fig 6), the average values were obtained for each repetition. For this index, as expected, there is an increasing trend with the growth of probability which implies that more nodes can transmit the message in the same time-slot. It also should be noted that the cases with 75 and 100 radii do not follow the highlighted patters: the reason why is because, as the radius and the probability increase, the possibility of having many nodes reached by the infection message in the first instant is high.

3 .3 Hierarchy of priorities

Thinking of a real system we have elaborated a hierarchy of priorities to try to understand which system was the most adherent to these characteristics:

1. High coverage percentage
2. Minimum coverage time
3. Minimum collision percentage

Considering runs of interest, shown in the graph 3a, we filter only those with a confidence interval on the mean above the threshold. We can rank them based on the coverage time, using Figure 5 and we see that an increase in probability is clearly beneficial for a system speed-up although with a diminishing return. We can observe that the confidence interval for $p = 0.3$, $r = 75$ collapses to the mean value, same as almost every run with $p = 0.15$ and thus given that it has the smallest coverage time between those, it is chosen as the best configuration for the previously explained metrics.

3 .4 Time evolution

Thanks to the data collected, we have the state of the system for each instant of time, and we are able to analyze the temporal evolution of the system. Considering the case with 200 nodes, we examine all four combinations of radius $(10 - 75)$ and probability $(0.15 - 0.85)$, plus the special configuration $p = 0.30$ and $r = 75$. Figure 7 is plotted on a linear time scale and shows the relative speed of completion for the different configuration. In particular, it is evident the negative impact given by a small probability of transmission, especially if coupled with a small radius. Unfortunately, this figure doesn't show confidence intervals, as they would clutter the graph too much, particularly in the most critical initial instants. Figure 8 tries to fix this problem by employing a logarithmic scale on the time axis. This allows us to better define differences between configurations, as the colored areas represent a 90% confidence interval. It is clear that the special configuration $p = 0.30$ and $r = 75$ is the fastest one between those that reach a complete coverage. Increasing the probability makes the system faster given that we increase the radius too: this is the difference between the configuration $p = 0.85$ $r = 75$, which is the fastest to reach an almost 90% coverage with a 90% confidence level, and the configuration $p = 0.85$ $r = 10$, which has a higher chance of representing an unconnected graph. Finally, if the problem imposes a small radius, it is clear that drastically reducing the probability (configuration $p = 0.15$ $r = 10$) guarantees a higher final coverage in regards of other runs with higher probability (configuration $p = 0.85$ $r = 10$), as we get much fewer collisions. Obviously we must accept the fact that this implies a much longer wait before reaching the final state.

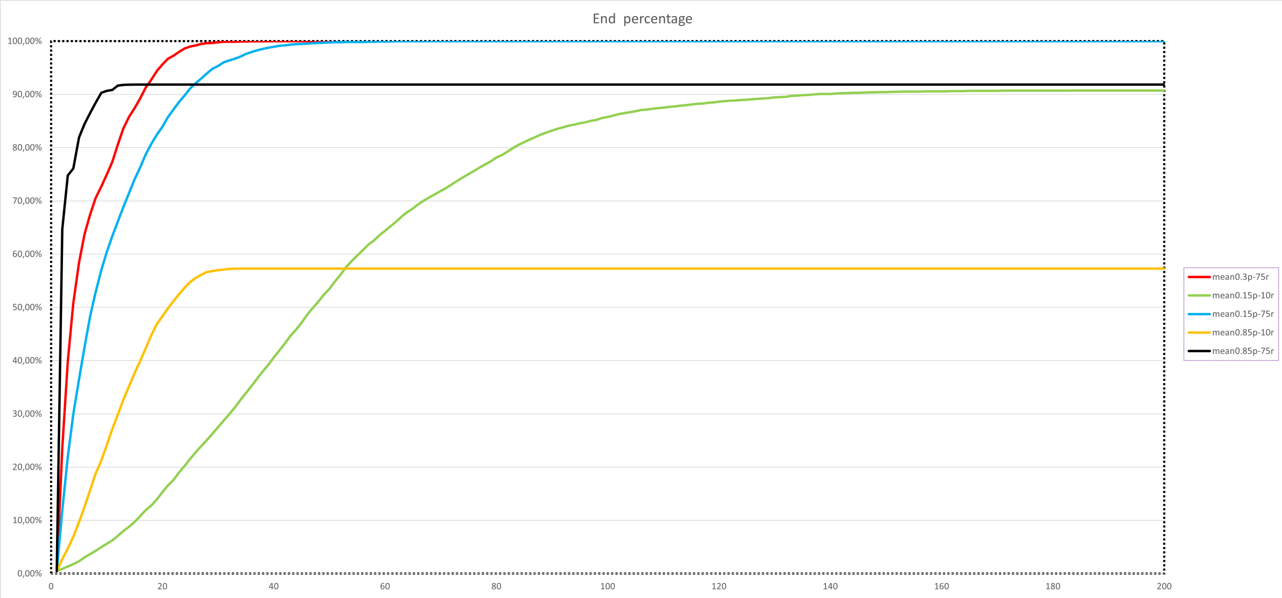


Figure 7: Mean done status

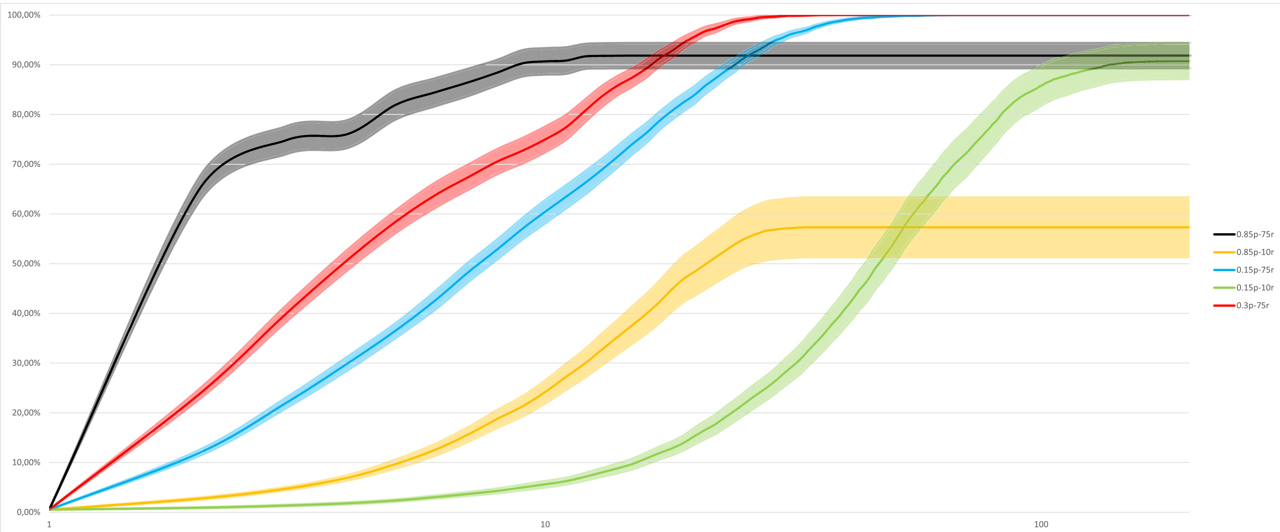


Figure 8: Done status on a logarithmic scale with 90% CI

3 .5 Factorial Analysis of ending time and ending coverage

We compute a 2^k factorial analysis to highlight the influence of radius r and probability p on two of our performance indexes, fitting the output variable to a bi-dimensional nonlinear regression model in the form of

$$y = q_0 + q_R \cdot r + q_P \cdot p + q_{RP} \cdot r \cdot p$$

Starting with the final node coverage, i.e. the fraction of nodes in the Done status at the end of simulation, extreme values are shown in Table 3 and the final influence in Table 4. It is important to notice that the extremes do not include a sending probability of 100% and radius of 100 because we decide to handle these as particular configurations that serve only to show the limit cases for the system.

Coverage percentage		Radius	
		10	**75**
Probability	**15**	0,9073	1
	85	0,5730	0,9183

Table 3: Extreme factor levels for ending coverage

	I	**Probability**	**Radius**	**Combined**	**Cov. Perc**	**SD_i**
	1	-1	-1	1	0,9073	0,0033
	1	1	-1	-1	0,5730	0,0765
	1	-1	1	-1	1,0000	0,0226
	1	1	1	1	0,9183	0,0047
4q	3,3986	-0,4159	0,4380	0,2526	Total	0,1072
q	0,8497	-0,1040	0,1095	0,0631		
4 q^2		0,0432	0,0480	0,0159		
Influence		0,4036	0,4476	0,1488		

Table 4: Influence of factors for ending coverage

We get a pretty balanced influence between radius and probability of about $\sim 42\%$ each, although we must notice that the term q_P is negative (-0.10), which implies a reduction of the final coverage with an increase in probability. Looking at Figure 6 we can clearly understand that this is primarily due to an increase in collisions, which naturally reflects the much lower coverage. Finally, we can assess that the interaction of both parameters is relatively small (14%), and thus we can study the system by just fixing one parameter and varying the other.

Regarding the completion time, the extreme values are shown in Table 5 and influences in Table 6

Coverage time		Radius	
		10	**75**
Probability	**15**	111,0152	47,6515
	85	21,4091	7,2576

Table 5: Extreme level of factors for time coverage

	I	**Probability**	**Radius**	**Combined**	**Cov. time**	**SD_i**
	1	-1	-1	1	111,0152	4119,3058
	1	1	-1	-1	21,4091	646,3921
	1	-1	1	-1	47,6515	0,6694
	1	1	1	1	7,2576	1566,2406
4q	187,3333	-130,0000	-77,5152	49,2121	Total	6332,6079
q	46,8333	-32,5000	-19,3788	12,3030		
4 q^2		4225,0000	1502,1497	605,4582		
Influence		0,6672	0,2372	0,0956		

Table 6: Influence of factors for time coverage

It is clear that having both negative coefficients ($q_P = -32.50$ and $q_R = -19.38$) is beneficial for a faster system termination (lower is better). The probability is also $\sim \times 3$ more effective at reducing the completion time, which is also stated by Figure 5 and is in line with what is expected: even if it means having more collisions, having a higher probability reduces the time a node waits in the Ready status. Again, the interaction of both parameters is negligible and allows us to study the effect of each parameter separately.

4 Model fitting

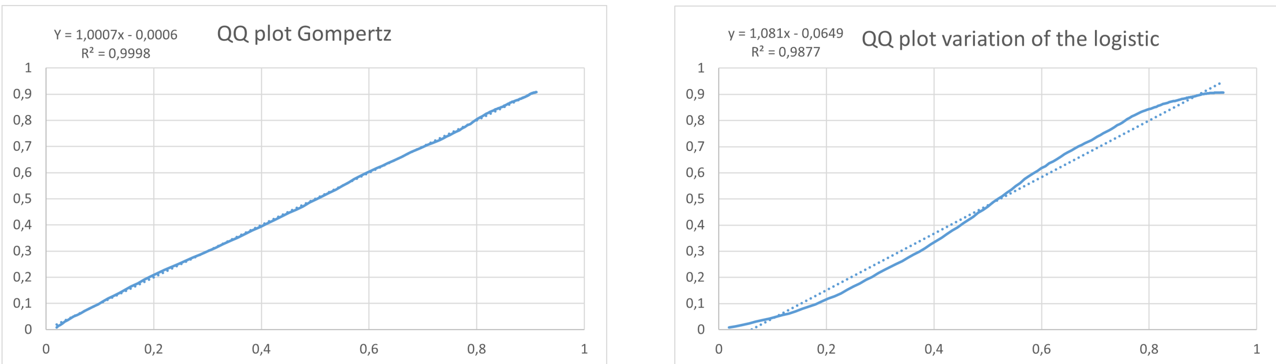
In the following section we try to find and determine a model to describe the evolution of the done status of the nodes as a function of time elapsed. Given the nature of the problem at hand, there are several options for models that might describe the evolution of the done status over time:

- Logistic model $Y = \frac{A}{1+e^{-Rt}}$
- Variation of the logistic model $Y = A\frac{1-e^{-Rt}}{1+e^{-Rt}}$
- Gompertz model $Y = Ae^{-Be^{-Rt}}$
- Asymptotic model $Y = A + Be^{-Rt}$

The next step is to find the most adapt for our case: by looking at the requirements and assumptions of each model, some of them need to be excluded. The first one is quickly discarded because it is more fit to describe the probability of success for a given case, even though the curves between the model and the time evolution of the system look similar, it would be wrong to use the equation improperly. The asymptotic model is discarded because its second derivative has no change in sign, this is not applicable to the system because, as show in Figure 7 the function can have a change in curvature.

Both Gompertz and a variation for the logistic model are used to represent "growth of population" dynamics and could adapt to the system case, but some checks need to be done before proceeding any further.

By performing a residual analysis the Gompertz model results to be the most suited for our samples i. e. for the configuration of $p = 0.15$ $r = 10$ the LSE value for Gompertz is 0,0027 meanwhile the one for the variation is 0,2984. Note that these values are very close to 0 since a spreadsheet solver was used to minimize the LSE (with the found values $A = 0.9117$ $B = 4.0959$ and $R = 0.0414$), a QQ plot for this configuration is shown below. In addition the use of the Gompertz model fits the logic of one of its typical case: examining diseases spread. Regarding the variation of the logistic model the value generated by the solver were $A = 0.9474$ and $R = 0.0264$



(a) Gompertz model (b) Variation of the logistic model

Figure 9: QQ plots for the models

The QQ plot confirms linear relationship between the values from our sample and the ones from the Gompertz model, this is also indicated by the value for R^2.

5 Testing the assumption for the Gompertz model

The next thing to do is to check the implicit assumption made to use the least square minimization. It is required that the residual are IID normal variables with null mean and constant standard deviation.

Firstly we check for IIDness.

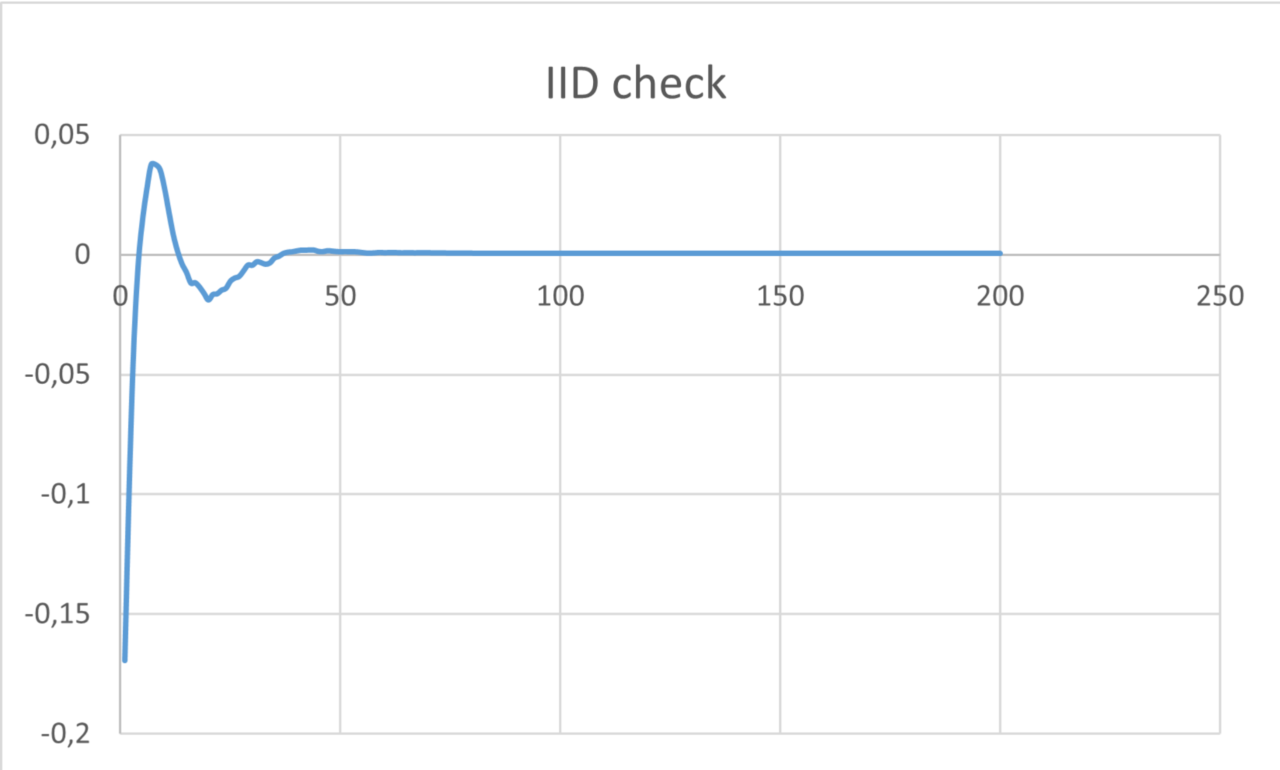


Figure 10: check for IIDness

It can be easily noted that there is in fact a pattern throughout the whole measurements.

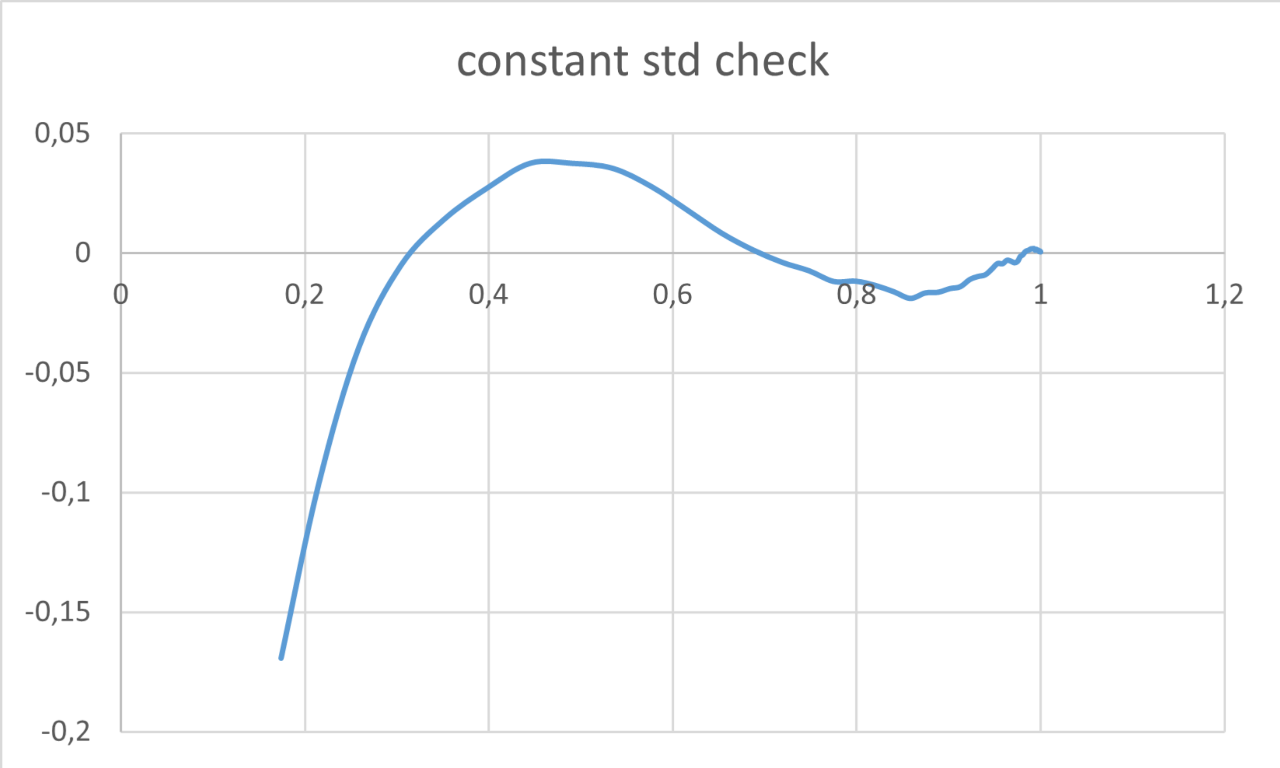


Figure 11: check for constant STD

Unfortunately a pattern can also be seen for the constant STD check.

This is bad news because, even though the Gompertz model seems to fit the case at hand, it does not pass the check for the implicit assumption to use the least squared method. Please note that in this section only the test for the Gompertz models are shown, but the variation of the logistic model was also tested, producing similar results

6 Varying node number

In this chapter we observe what happens when the number of nodes in the system varies. We summarize the value of coverage percentage and the coverage time in the table below.

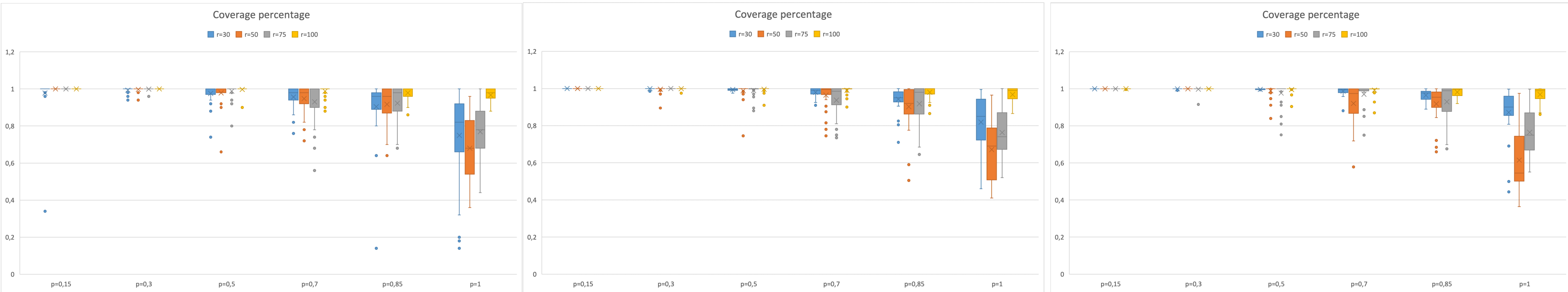
		Coverage percentage						Coverage time					
r	n	p=0,15	p=0,3	p=0,5	p=0,7	p=0,85	p=1	p=0,15	p=0,3	p=0,5	p=0,7	p=0,85	p=1
10	50	0,024482	0,024482	0,024482	0,025843	0,025665	0,025665	17,16667	6,984848	4,560606	3,469697	2,80303	2,651515
10	100	0,02159	0,024533	0,017109	0,016526	0,018371	0,02299	41,37879	18,92424	11,71212	8,318182	6,590909	5,166667
10	200	0,747209	0,7723	0,588076	0,468021	0,304903	0,197414	111,0152	61,86364	37,83333	28,01515	21,40909	16,89394
10	500	0,998189	0,994068	0,978308	0,945551	0,906997	0,83855	65,62121	38,89394	28,43939	24,92424	21,89394	21,07576
10	700	0,999567	0,996217	0,98109	0,963497	0,93384	0,874377	62,01515	37,80303	27,95455	24,4697	21,92424	21,04545
30	50	0,86282	0,977003	0,92532	0,89483	0,74599	0,500591	33,71212	18,77273	13,98485	10,65152	9,621212	6,469697
30	100	1	0,98707	0,940233	0,912344	0,854607	0,574112	35,37879	23,25758	16,34848	13,80303	11,28788	7,833333
30	200	1	0,996414	0,987382	0,955466	0,881255	0,677805	45,13636	26,01515	20,56061	17,19697	13,68182	9,924242
30	500	1	0,999182	0,989961	0,956269	0,913185	0,695982	58,34848	40,22727	30,59091	24,22727	19,65152	13,13636
30	700	1	0,997563	0,990985	0,964937	0,935377	0,732669	64,95455	44,98485	34,65152	27,92424	21,92424	16,89394
50	50	1	0,985655	0,917359	0,875088	0,814974	0,499584	30,5303	16,62121	12,28788	9,136364	7,378788	3,439394
50	100	1	0,997956	0,950017	0,888699	0,7652	0,454022	35,56061	24,25758	16,4697	12,59091	8,560606	3,590909
50	200	1	0,976742	0,942634	0,898068	0,788035	0,500295	49,04545	29,59091	21,31818	15,65152	11,10606	3,80303
50	500	1	0,995504	0,985126	0,913678	0,749905	0,463861	67,71212	45,5303	30,5303	21,04545	13,89394	4,863636
50	700	1	1	0,957128	0,814391	0,824465	0,453678	77,07576	48,59091	33,37879	19,98485	15,5	5,19697
75	50	1	0,991825	0,94982	0,817595	0,823015	0,632061	29,65152	16,01515	10,83333	6,348485	4,893939	1,5
75	100	1	1	0,921545	0,896459	0,819265	0,649472	35,25758	22,40909	13,40909	9,106061	5,984848	1,5
75	200	1	1	0,96036	0,844924	0,806358	0,638544	47,65152	25,31818	16,77273	10,5303	7,257576	1,5
75	500	1	0,972766	0,915629	0,863574	0,760205	0,637793	61,28788	35,34848	22,56061	13,28788	8,045455	1,5
75	700	1	0,982774	0,91658	0,909074	0,828044	0,642411	66,68182	38,01515	23,37879	14,25758	9,621212	1,5
100	50	1	1	0,979562	0,95875	0,942408	0,926149	28,5	13,25758	8,227273	5,015152	3,378788	1,5
100	100	1	0,979562	0,984599	0,96916	0,954525	0,937334	31,62121	17,68182	9,893939	5,893939	4,378788	1,5
100	200	1	0,99489	0,979734	0,967563	0,944922	0,928947	41,01515	18,28788	11,04545	7,075758	4,651515	1,5
100	500	1	0,999591	0,979969	0,964415	0,957983	0,929607	48,04545	24,71212	13,28788	8,106061	5,772727	1,5
100	700	0,999124	0,99854	0,9784	0,967052	0,956693	0,931083	49,43939	24,71212	13,43939	9,227273	5,378788	1,5

Table 7: Coverage percentage and coverage time for different scenarios

It is interesting to notice that lower radii are better suited at higher number of nodes for a better coverage percentage. This happens because the area in which the nodes are distributed does not change so the node density increases as the number of neighbors; this increases the chances for a fully connected graph. Another thing to notice is that, for the completing time, the values increase with the number of nodes, but they become much worse when the sending probability is lower. This divides each rectangle (one for each group of radii) in two triangles, the lower left with the worse times and the upper right with the best ones and, with a growing radius, the last one gets bigger. Finally, we can see that lower probabilities tend to perform better than higher ones for coverage percentage despite coverage time, independently from the number of nodes and radius. The only exceptions are the extreme cases of really low radii and number of nodes ($r = 10$ with $n = 50 - 100 - 200$ and $r = 30$ with $n = 50$). The confidence intervals for these mean values are provided in the appendix.

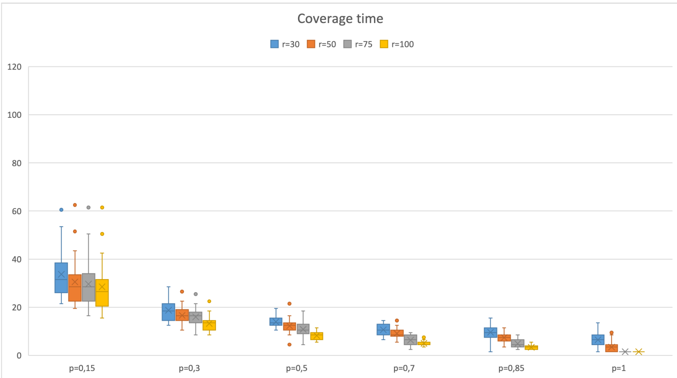
6 .1 50 and 700 cases

The extreme values we consider for node number are 50 and 700. We now want to analyze the performance indexes for these configurations, to discover any similarities and/or differences with the case at 200. Note that for readability, the configuration of $r = 10$ for 700 nodes is not shown, even if it should not be excluded since its coverage percentage is within acceptable range. As can be seen in figure 12, by varying the number of nodes, the visible trends for the case of 200 nodes as highlighted in section 3 .2 seem to be confirmed. As regards the coverage time at low probabilities, an increase is noted, as expected, as the number of nodes increases. It is also visible how as the number of nodes increases, and therefore their density in the plane, the number of collisions increases, even for very low radii and probabilities. However, it must be specified that a high number of collisions does not necessarily imply a worse percentage of coverage. It should also be noted that an increase in node density has two different effects on data variability. Firstly, an increase in variance for coverage time caused by a higher percentage of collisions. Secondly, a decrease in variance for coverage percentage due to the fact that the same amount of nodes that do not receive the infection message correctly accounts for a minor portion of the total coverage.

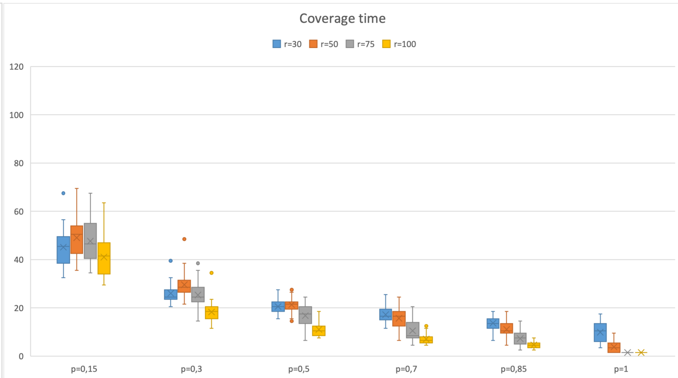


(a) 50 Nodes (b) 200 Nodes (c) 700 Nodes

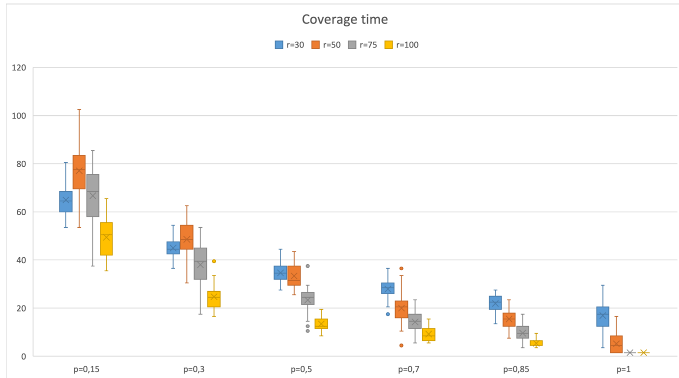
Figure 12: Coverage percentage



(a) 50 Nodes

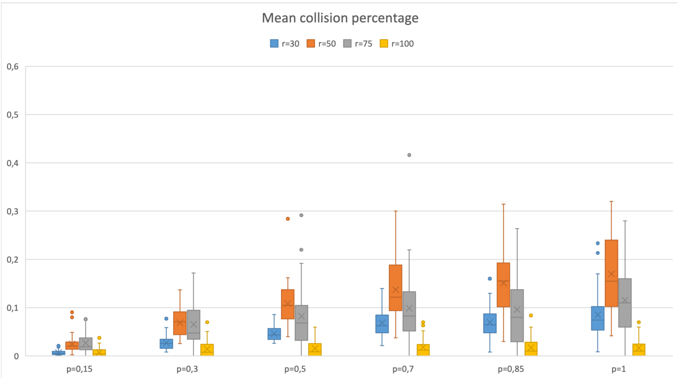


(b) 200 Nodes

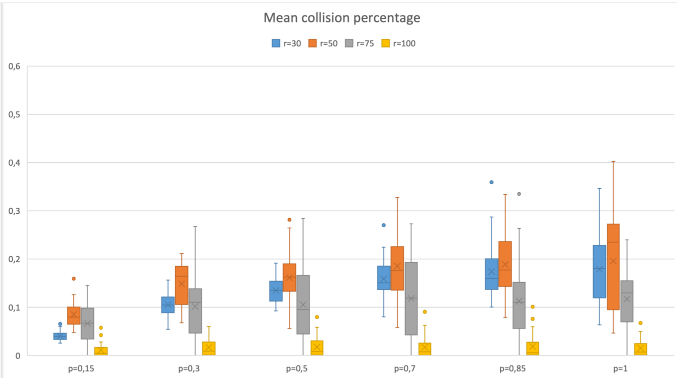


(c) 700 Nodes

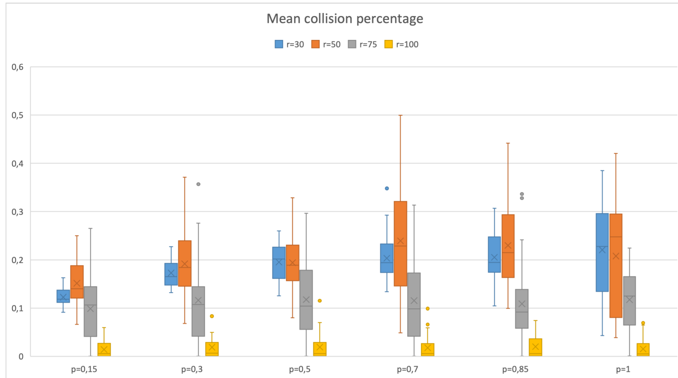
Figure 13: Coverage time



(a) 50 Nodes



(b) 200 Nodes



(c) 700 Nodes

Figure 14: Mean collision percentage

7 Factorial analysis on different number of nodes

The purpose of this section is to confront the results of the factorial analysis for various models with different number of nodes, since the process for obtaining these results is the same explained in section 4 , we shall skip directly to the values for influences.

7 .1 Influence on coverage rate and time

We first analyze the result for the coverage percentage and the coverage time.

Number of nodes	Probability	Radius	Combined
50	*0,0021*	0,9961	*0,0018*
100	*0,0054*	0,9939	*0,0007*
200	*0,4036*	0,4476	0,1488
500	*0,9048*	*0,0464*	*0,0488*
700	*0,9580*	*0,0208*	*0,0212*

Table 8: Influences for coverage percentage

Number of nodes	Probability	Radius	Combined
50	*0,8269*	0,1148	*0,0584*
100	*0,9819*	*0,0108*	*0,0073*
200	*0,6672*	*0,2372*	0,0956
500	*0,9571*	*0,0336*	*0,0092*
700	*0,9646*	*0,0060*	*0,0294*

Table 9: Influences for coverage time

To draw easier conclusions, in the following section, these results are plotted in a scatter plot with lines. Note that the values in italic are referred to influence of factors with a negative contribution meaning that a higher value for that specific factor produces a lower output, however this is not a bad thing because in the case of the coverage time the objective is to minimize it.

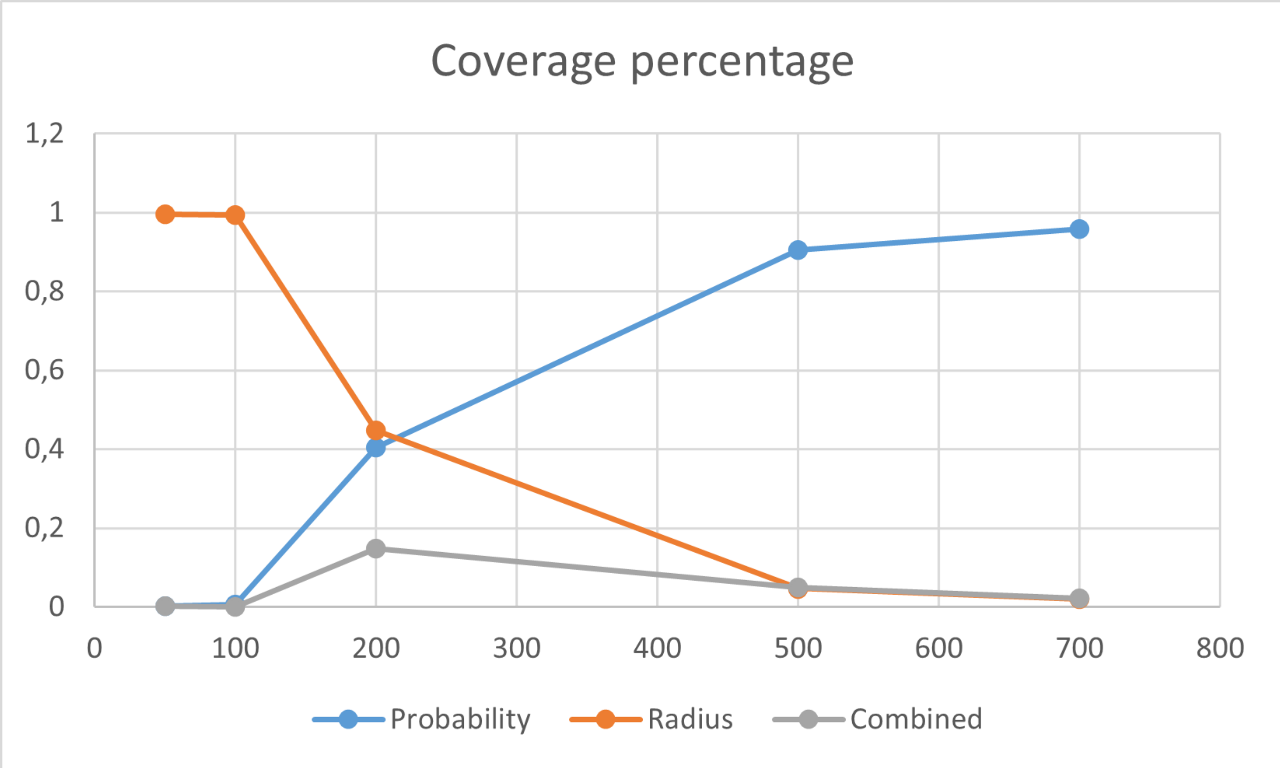


Figure 15: Coverage percentage at different nodes

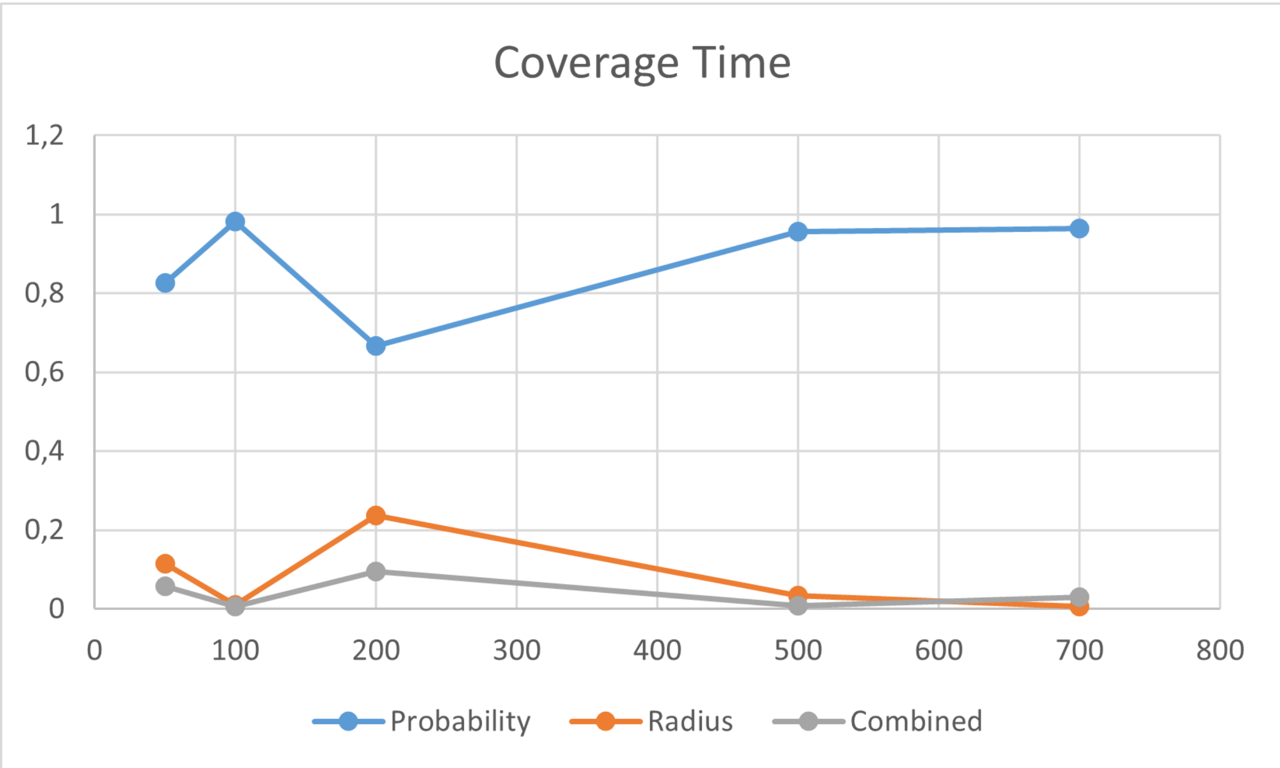


Figure 16: Coverage time at different nodes

Let's start with figure 15. It is easy to understand that a low density the radius is far more critical then the probability since with low radius it is likely that many nodes are isolated or do not get a second "chance" after receiving a collision. At a high number of nodes, the contrary happens: the reason of this behavior is that even with low radius, each node has a lot of neighbors so it is far more important for them to avoid collisions. This is achieved by a low probability as seen in the boxplot and as indicated in figure 12.

The coverage time is more complex to fully understand: with a high number of nodes it is far more important for nodes to quickly transmit. However with a low density the behavior for the influences is more chaotic, this happens because, since this analysis takes in account only the coverage time, some runs end immediately with a high probability and low radius. However these configurations result in a very low coverage percentage. With 200 nodes a critical point is reached, the reason behind is that this configuration has a high enough density for the runs to not immediately terminate as in the previous cases. This still determines a high influence for the probability factor, but still gains from a higher radius so that more nodes are reached at each transmission. Note that almost all of the values for influence in table 9 are in italics, therefore negative: this make sense because our objective is to diminish the coverage time overall.

8 Conclusions

With this analysis we've shown the varying influences that each parameter has on system performance.

We can state that in an ideal scenario, having a very large radius that allows every node to connect with almost every other part of the graph is enough to respect every performance index that we considered (section 1 .2). The system is therefore fast and reliably delivers the message across the entire floorplan (section 3 .4). A high probability of transmission is therefore encouraged for a faster completion.

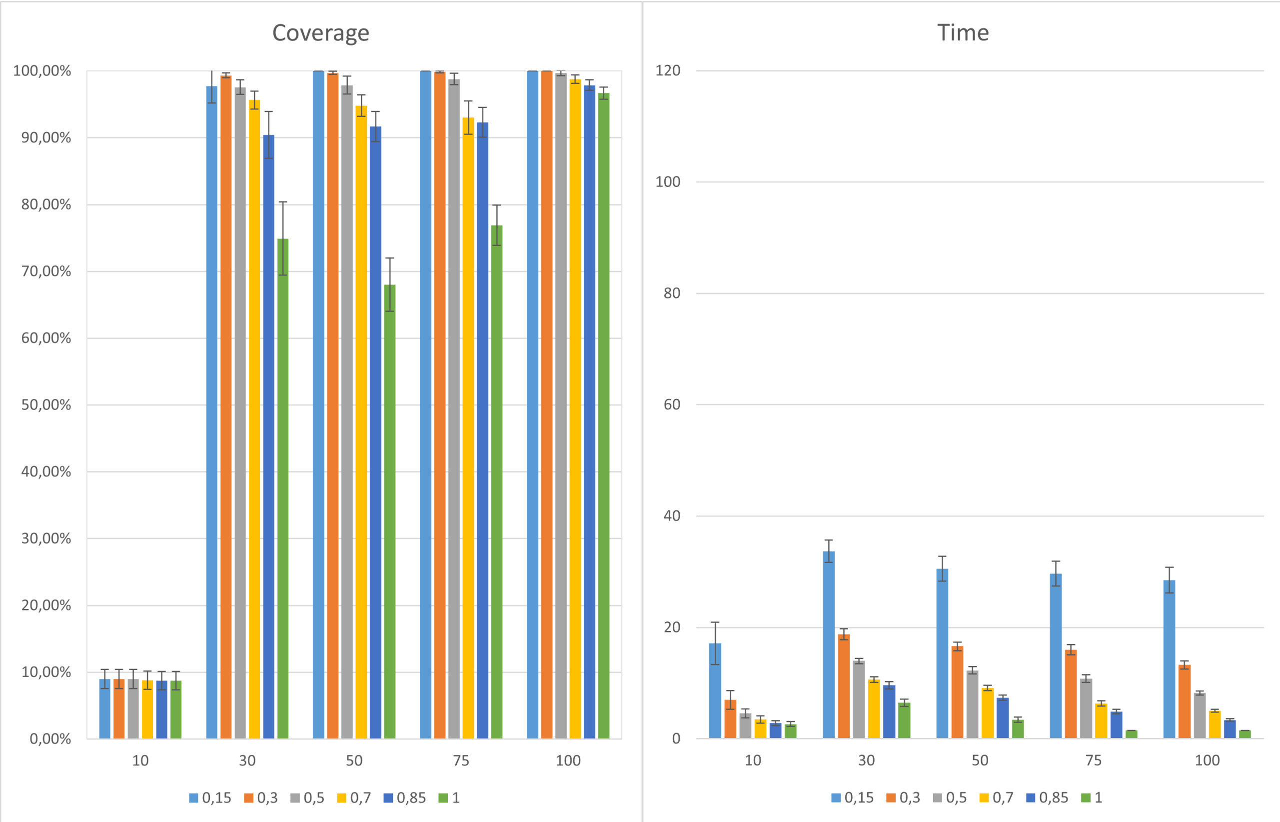
Unfortunately, real systems often cannot satisfy this unrealistic assumption: for physical reasons, often related to system power and distances, nodes are limited by a very narrow view of the system, that can often lead to a higher chance of dealing with a lightly connected or even unconnected graph. In this cases, system engineers should tune down the probability threshold below 50%, which drastically reduces collisions and increases the possibility to reach all nodes (Fig. 12). Assuming we require to complete with a total coverage percentage, runs with lower probability and a high enough radius to avoid unconnected graph cases perform the best (Fig. 3). In case we prefer a faster system response in favor of a lower final coverage, it is possible to employ a higher probability threshold, with a global trend as shown in section 6 .1.

Regarding the assumptions made for this analysis (section 1 .3) it would be possible to extend the range of other floorplan configurations, especially the condition of non-uniformly distributed nodes. This would allow the modelling of sub-networks inside the whole graph, with the added effect of certain nodes having the role of a 'bridge' between groups and probably a higher chance of resulting in a bottleneck.

This analysis explored the variation of multiple parameters and we often choose to consider very large extremes. For specific cases, it would be much beneficial to focus the research on a narrower range of values, increasing the resolution for each parameter of interest.

A Appendix

A.1 50 Nodes

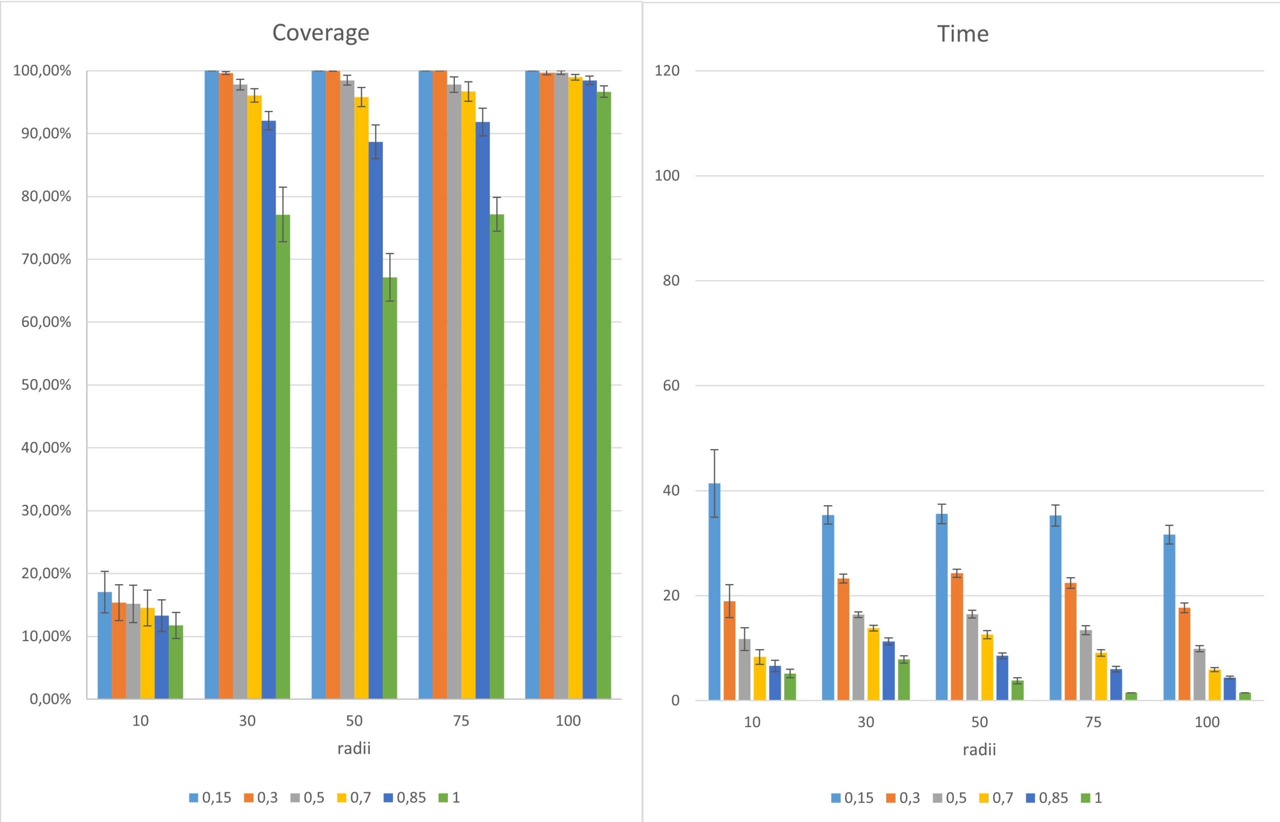


(a) Coverage percentage

(b) Coverage time

Figure 17: 50 Nodes confidence interval (90 %)

A.2 100 Nodes

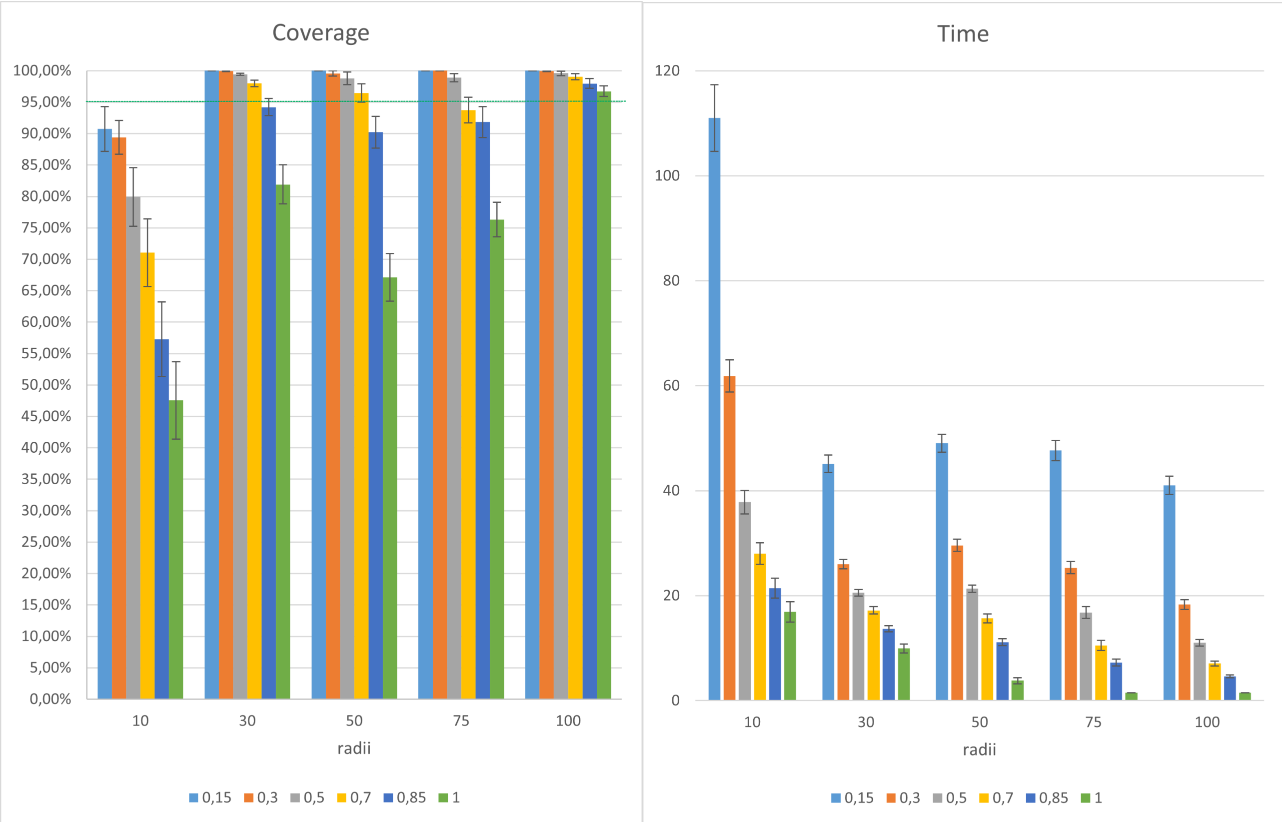


(a) Coverage percentage

(b) Coverage time

Figure 18: 100 Nodes confidence interval (90 %)

A.3 200 Nodes

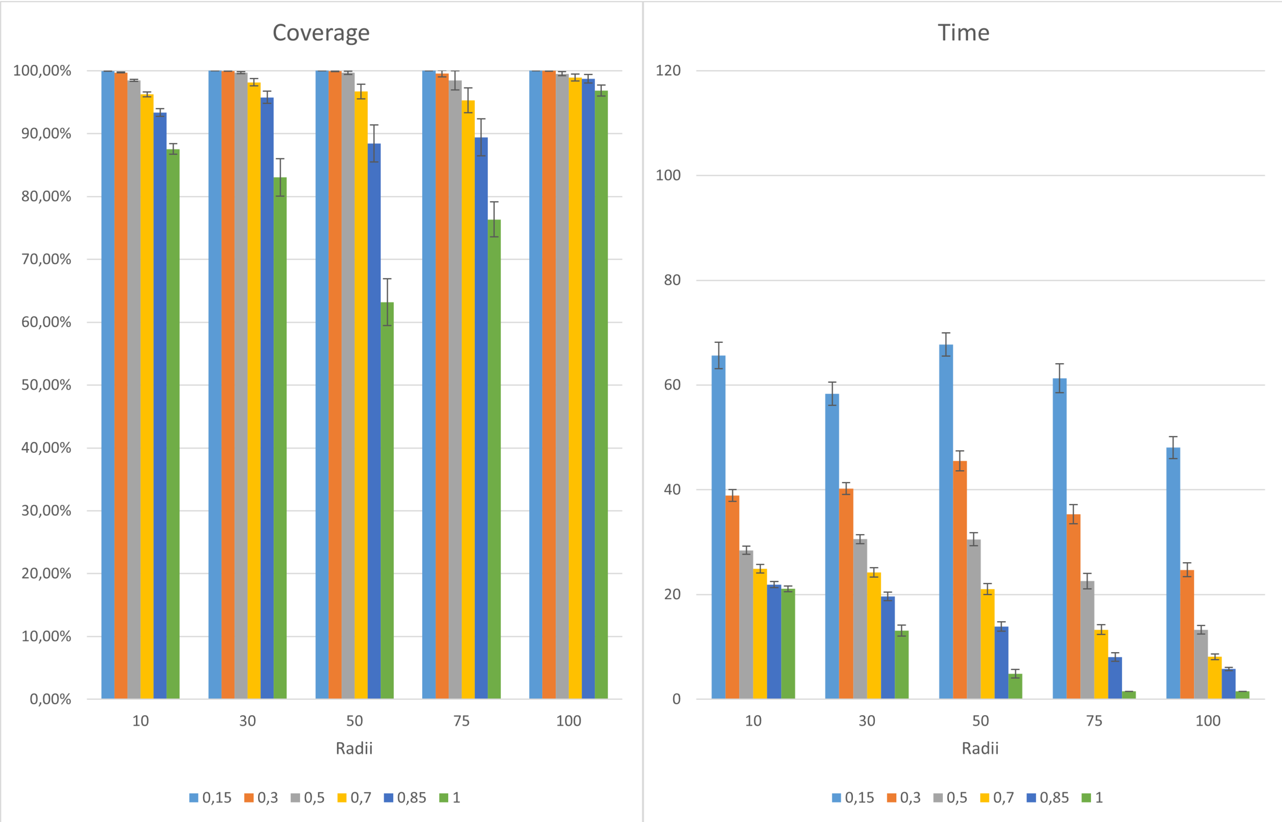


(a) Coverage percentage

(b) Coverage time

Figure 19: 200 Nodes confidence interval (90 %)

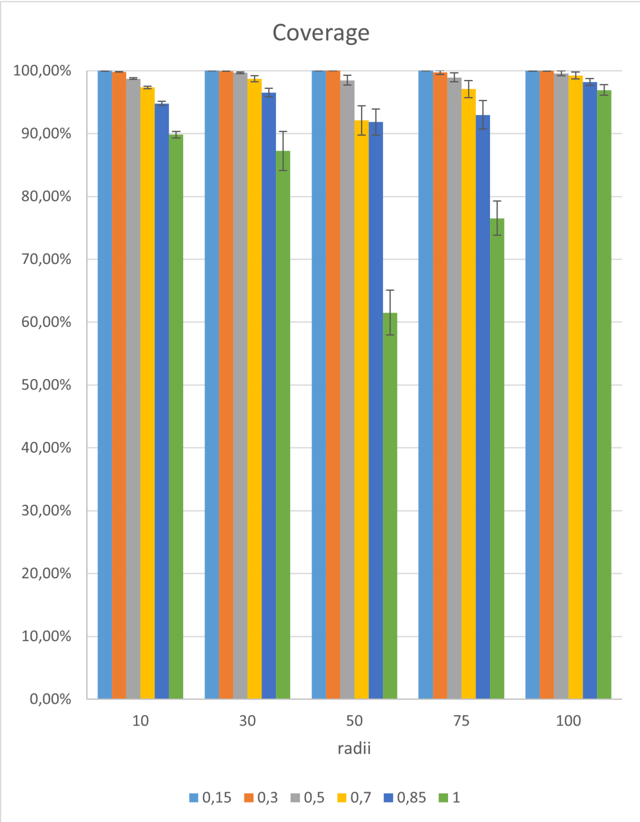
A.4 500 Nodes



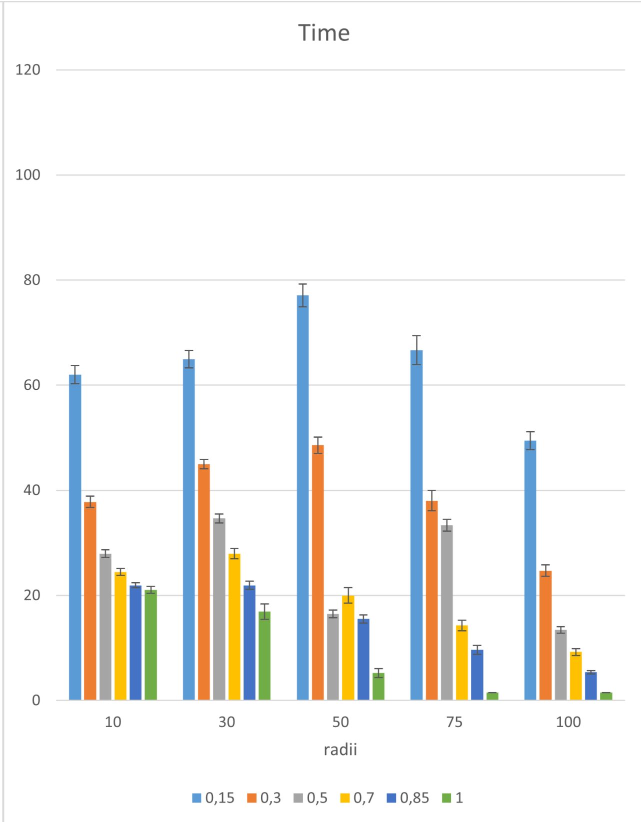
(a) Coverage percentage (b) Coverage time

Figure 20: 500 Nodes confidence interval (90 %)

A.5 700 Nodes



(a) Coverage percentage



(b) Coverage time

Figure 21: 700 Nodes confidence interval (90 %)