Ατομική Διπλωματική Εργασία

PERFORMANCE EVALUATION OF THE EXPLORE AND EXPLOIT ALGORITHM IN EMERGENCY RESPONSE NETWORKS

Ismini Evagorou

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2017

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

PERFORMANCE EVALUATION OF THE EXPLORE AND EXPLOIT ALGORITHM IN EMERGENCY RESPONSE NETWORKS

Ismini Evagorou

Επιβλέπων Καθηγητής
Andreas Pitsillides

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2017

Acknowledgements

First I wish to express my sincere thanks to Prof. Andreas Pitsillides for believing in me and providing me with the opportunity to work on this project, which has given me the privilege to widen my knowledge to the core areas of research. Furthermore, I would like to show my appreciation for his valuable guidance and encouragement provided in many stages for the development of the current study. He was always willing to help and offer estimable advices.

Secondly, I would like to take the opportunity to express my profound gratitude to Dr. Panayiotis Kolios for sharing his expertise and his pearls of wisdom with me during this research. Panayiotis advices and suggestions were of immense help throughout my study.

Last but not least, I would like to acknowledge my family and friends for their moral and emotional support and patience along my way.

Abstract

The past years we have experienced many natural disasters which caused the death or injury of several people. We have made major efforts to deal with these kinds of incidents. Our intention is to cure the injured people and attend to the stranded survivors faster and thus decrease the loss of human lives.

The purpose of this thesis is the implementation of a system which will help with the localization of the survivors and avoid the unwittingly human losses. Based on this system the first aid responders will omit the widespread searches of stranded survivors that leads to deadlocks and they succor them instead of wasting time to locate them.

Thankfully, at current times people of all ages have smart phones in their possession. We can use the rapid development of technologies in our best interest. The whole idea is the smartphone to send a message on behalf of their holder in order to be read by their rescuers. The messages will comprise the location of each survivor so the response unit will know their position. The mobile devices will form an ad-hoc network when the infrastructure-based communication system is collapsed. The smartphone of each survivor will communicate with other reachable smartphone (devices in their range) in order to disseminate sufficiently their message throughout the network.

The creation and preservation of the network has to be made with the right algorithms. The network must stay "alive" for enough time, to forward the messages to the rescuers when they arrive. The devices must not drain out of battery in short period of time because the network will not expand enough. As a result of this some survivors may stay unrevealed and lead the rescuers to bypass them.

In brief, when a disaster strikes, the devices of the survivors will create an adhoc network dynamically. The mobile devices will forward their location through this network using a high-performance dissemination policy in order to ensure the longevity of the network. In this way, the first aid responders receive the messages from a representative node of the network, so that they can locate the survivors based on the received messages and provide care to the wounded/stranded survivors. And above system does not add considerable cost for building new dedicated infrastructure (negligible in comparison to a dedicated Disaster Recovery Network).

Contents

Tables

Images

Chapter 1

Introduction

1.1 Problem Definition

The humanity has experienced many strikes of natural disasters in the past years. This kind of disasters are unpleasant and unforeseeable. Except from the consequences in nature, there are consequences to humanity too. After a disaster strike, there are survivors which are injured, trapped or both in disperse locations. The rescuers have to discover these locations and help them.

A lot of time has been wasted in order to locate stranded survivors. Even though the conditions and the lack of human resources are not helpful, by this time, we should find out a way to locate and rescue the stranded survivors faster and more effectively. Rescuers must track down the injured or trapped persons before is too late for them. Some survivors may not be discovered and end up dead.

Bottom line natural disasters have caused the deaths and the injury of many people and since these situations are unavoidable, we need a more effective system to confront them.

1.2 Study Purpose

Whenever a tragedy happens, as a natural disaster, people could be anywhere. These events are unpredictable most of the times and people are taken off guard. We need a

common and everyday object that people may have on them every hour of a day to use it as our advantage.

The purpose of this study is to benefit from the rapid development of technology, in order to locate the stranded survivors of a disaster, faster and efficiently. Since at these times, people of all ages have a smartphone in their possession, we came up with the idea of creating an ad-hoc networks using their smart phones. Specifically, the creation of an ad-hoc network from the mobile devices of the survivors. Each phone will disseminate the location of his holder through the network by sending a message to its reachable neighbors (nodes in an adhoc network). When the rescuers locate one of the survivors they will be able to read the help-requests of the others and find their location without effort. In so doing, the time of localization of the survivors will be significantly reduced.

Nevertheless, the dissemination policy of the help-requests must fulfill specific requirements, such as the longevity of the network. When the rescuers reach the district of the network they must be able to read the requests for this reason the network must stay active for a reasonable period of time. Also, the dissemination algorithm must send the request across disperse locations to increase the probabilities of tracing each survivor. Although the implementation must provide a highly efficient data propagation to save the energy of the devices.

To conclude, when a catastrophe happens and the infrastructure-based communications systems are destroyed, the mobile devices of the survivors will form an ad-hoc network and disseminate help-requests. By creating a self-sufficient dissemination algorithm, the survivors are reassurance that they will be found.

1.3 Review

The current study consists of five chapters.

In chapter 2, are described the main background meanings for the ad-hoc networks. Additionally, are explained some already existing operations that are included in the implementation of the algorithm.

In the chapter 3 is explained the algorithm and operation of the dissemination strategy that is used, its implementation and the reason that is chosen. Furthermore, the algorithm functions are analyzed and explained in detail.

For the fourth chapter, the scenarios that were made to evaluate the efficiency of the algorithm are explained. Also the results of the scenarios are presented and at the end the metrics are compared.

In the chapter 5, a conclusion about the purpose and the results are presented in order to check if the expected results were accomplished, as well as future directions.

Chapter 2

Background

After a disaster strike, a network is instantly created from the mobile terminals and there are originated to disseminate their requests. Moreover, the forwarding of messages should be intelligent. Due to the efficiency of the algorithm we create heuristics to make forwarding decisions. These metrics are Local Centrality (LC) and Eigenvector Centrality (EC). The Local Centrality of each node is analogous with the battery remains of the device and the centrality of the node in the network.
Using the metrics each terminal can determine it role in the network. The role that is taken from the devices will bound their forward policy. There are four different kinds of roles, cluster heads, bridge nodes, leaf nodes and intermediate nodes. The cluster heads nodes, are the leaders of a region. Also, there is the role of Bridge nodes, whose are the nodes that belong in two regions and as a result connect two cluster heads. The bridge nodes can determine their role with signaling process. Moreover, leaf nodes are the nodes that are at the edge of the network and have the lowest values. Besides these

roles, there are the intermediate nodes, the nodes that are members in one cluster but they are neither cluster head nor leaf nodes. Some of the immediate neighbors of intermediate nodes have higher LC value and others have lower. Once the nodes mark off their role in the network, we can see that the nodes are organizing into groups. In every group, each node takes a particular role.

Based on the values of LC, EC and the role of each node in the network we can devise highly- effective algorithms, which include smart propagation strategies.

Since the purpose of the network creation is the facilitation of the response units, we need to assurance the network is going to be active until the time they arrive. In order to achieve the network longevity, we define two different modes for the nodes, active mode and sleep mode. Each node calculates its active period time up to a value and during a cycle, which is a fixed time period, is active for that time, and for the rest of the cycle is in sleep mode. As a result of the modes, the devices preserve their containing energy during the sleep mode and the period that the network is active is extended.

2.1 Local Centrality

$$x_i = e_i + \frac{1}{\|x_{\mathcal{Z}(i)}\|_2} \sum_{j \in \mathcal{Z}(i)} A_{ij} x_j, \ \ \forall i \in \mathcal{N}$$

Image 2.1 LC equation

i: the node that LC value is calculated for

x: the LC value

e: the battery inventory of the node

N: all nodes of the network

A: a binary value, the value equals with 1 if nodes i and j can communicate otherwise the value is 0

z(i): all the nodes that can reach node i and send him their requests.

In order to calculate the LC value, a node collects all the LC values of his direct neighbors and then compute his new LC value. The metric takes into consideration

other LC values from one hop away, but these values considers their one hop away neighbor values. So recursively the LC value of each nodes considers values from one, two, three or even more hops away.

The Local Centrality value help each node to evaluate his importance in the network. The estimation is using the remaining battery of the device and the centrality of the node in the network. The nodes with the higher LC values have stronger roles in the network than the nodes with lower values because their survivability probabilities are higher.

The network splits into clusters, using the local centrality values. Each cluster has a leader. The leaders of the groups are the nodes with the highest local centrality values, consequently the cluster heads.

2.2 Eigenvector Centrality (EC)

$$x_i = \Phi_i + E_i + \frac{1}{\|x_{\mathcal{Z}(i)}\|_2} \sum_{j \in \mathcal{Z}(i)} A_{ij} x_j, \ \forall i \in \mathcal{N}$$

Image 2.2 EC equation

i: the node that EC value is calculated for

x: the EC value

e: the battery inventory of the node

N: all nodes of the network

A: a binary value, the value equals with 1 if nodes i and j can communicate otherwise the value is 0

z(i): all the nodes that can reach node i and send him their requests.

Φ: a value between 3500 and 4500 if a node is bridge else the value is zero.

The EC value is calculated likewise the LC value. The use of this metric is to separate the bridge nodes from the other nodes of the network. Specifically, to distinct the leaf nodes that belongs to two clusters from the leaf nodes that belongs to one cluster. To accomplish the separation of the nodes, we need the Φ value, which is a big number we add to the EC metric of bridge nodes.

The Eigenvector centrality is used to help the cluster heads to forward the messages to other regions of the network. To propagate a request to other clusters, the request must send downwards to a linking node and then upwards to another cluster head. So, the separation of the leaf nodes is done to know in which leaf node should the request forwarded to in order to end up in others clusters. The technique which is used for the separation is called signaling.
The main idea is to send the message to a bridge node which is related with two regions and when the bridge receives the message it can be forwarded towards the other clusters.

2.3 Nodes

As mentioned before, each node is associated with one role so that the dissemination policy becomes more methodical. To determine the role of each node in the network you use the LC value. According to the value of the node and the values of his immediate neighbors, each node individually figures out his role in the network. As time passes the nodes may change roles because of the battery losses or the decrement of the local centrality values of their neighbors. Because of this, the network is not static. Remarkable observation is that the role of the nodes depends on the remaining battery capacity of the device and the centrality of it in the network.

2.3.1 Cluster Head

A node can become cluster head by checking if its LC value is higher from all the LC values of his neighbors. If a node is cluster head that means it is in charge of a cluster. Accordingly, it is responsible for gathering the messages of all the nodes that are members in its cluster and then reroute them to other clusters. Since is the node with the higher LC value it is more probabilistic to have the resources and stay alive long enough to fulfill this job.

2.3.2 Leaf Nodes

A node can determine if it is leaf, by comparing it LC value with the advertised LC values of its neighbors and it has the lowest. Usually leaf nodes barely have residual battery capacity or they have minimally number of neighbors. Additionally, a leaf node can be associated with more than one clusters and change it role. Usually a simple leaf node has only to forward it location to the nearest cluster head.

2.3.3 Bridge Nodes

Bridge nodes have the lowest LC value compared to all of their neighbors like leaf nodes. The particularity that make bridge nodes special is that they are members in more than one clusters. The method to distinguish the bridge nodes is the signaling. Basically, bridge nodes are the leaf nodes that belongs to more than one clusters. So, they can help to propagate the messages from one cluster to another. The responsibility of the bridge nodes is to link the clusters and transmit the requests from one cluster head to another.

2.3.4 Intermediate Nodes

A node is defined as intermediate, if some of the LC values of its neighbors are higher than it value, and some others are lower than it. Intermediate nodes are the nodes that connect the cluster heads with other nodes. They are forwarding towards the cluster head the requests of nodes that are not immediate connected with them, but they belong to its cluster. Moreover, they transmitting the requests from the cluster heads to bridge nodes. Basically are the nodes between the cluster heads and the bridge or leaf nodes.

2.4 Cycles

The expectations of the network are longevity and smart dissemination of the help-requests in order to end up to a reasonable percentage of cluster heads. If we emphasis on the longevity, we need to let the devices unused for some periods of time resulting in

extended life period of the device and consequently of the network. So, we defined the active time period and the sleep time period of nodes. Each node will compute a value up to cycle value in minutes, which is going to be the active time period of it. The rest of the cycle the node is going to be in sleep mode. As a result, the consuming of the device power during the sleep time period, is diminish in the least. The cycles of nodes are necessary for the network longevity even though the propagation of the requests is going to be less efficient.

2.4.1 Sleep Mode

During the time period that the node is in sleep mode, the device is inactive. The other devices are not able to detect it. Therefore, the node cannot contribute to the network operations. In particular, the device is not capable to send, receive or forward any request. The intention of the sleep mode is the maintaining of the residual battery capacity, which is accomplished with minimum losses.

2.4.2 Active Mode

The time interval that the nodes are in active mode, a device can communicate only with reachable mobile terminals. Which are the devices that are positioned in it detection zone and are in active mode the specific time. In addition, a mobile terminal can receive or send a request during it active cycle. Even though these devices can communicate, the request can be received only from nodes that do not already have it.

2.5 Matlab

Matlab is a matrix based programming language that relies it tasks on mathematics. Arrays are used to control the flow of values and manage importing and exporting data. Further matlab gives you the opportunity to create interactive and graphical user interface and write programs which can communicate with other programming languages. Also it includes commands which are utilized for plotting functions and

data. Additionally, is able to represent two-dimensional and three-dimensional graphics.

Matlab is widely used for academic research and algorithm implementations. For this study is the ideal option since it has forceful graphic tools and is facile to use for data analyzing. In order to optimize the algorithm, we have created a simulation of the network. A graph of nodes that are randomly positioned is formed in the simulation and the nodes start to send their message through the graph. Consequently, we can observe the path of each message. Moreover, we can create graphics about battery consumption, average path length and other metrics which is helpful for collating different scenarios.

Chapter 3

Proposed dissemination strategy algorithm operation and implementation

---

For the current thesis is implemented a simulation of the network. As explained before, the moment of the network creation the nodes (devices) start to figure up their local centrality values. Afterwards, using the LC value, the nodes are labeled with one role each. Subsequently, the nodes estimate their active duration for a cycle and they start the dissemination of the requests. To produce the desirable result, we did some speculations that help in more effective analysis. We made the assumption that the cycles of nodes(device) starts exactly the same second and that there are not message losses.

Is also important to mention that after a while, the nodes would know more than one message. Also, the nodes can send each message once, except if a better candidate came up later. There are different occasions that may have as aftermath a better candidate. Firstly, the best candidate is in sleep mode the moment the custodian is sending the request, so it ignores it existence and the message is send to other candidate.

When the cycles of the custodian and the best candidate get synchronized, the message is resent to it. Secondly, whilst the network is active, the battery of nodes is diminishing. Consequently, the LC and EC values are changing since the values are based on the remaining battery of the device**.** As a result, the values of certain nodes may end up higher than the current value of the previous receiver or the roles of the nodes might change. Additionally, an operation is included in the algorithm, which alters the EC values and helps to disseminate the requests towards more than one bridge nodes.

In order to fully understand the routing process, we can divide it into phases

A) select the source message from the phone database to forward it

B) find the neighbors which are available to read the message

C) select the best candidate to send the source message

D) set the message flag according to your role and the current value of the flag

E) send the message

We are going to give detailed explanation of the routing process for all the roles below. Moreover, there are explanations about additional alternations to the algorithm.

The implementation script of the network simulation is written in MATLAB, in order to create graphs and metrics to bring into comparison from different scenarios.

3.1 Signaling

The bridge nodes may not be directly connected with cluster heads, so we need to figure out a way to determine which leaf nodes are connected with two clusters. The cause of signaling process is to distinct which leaf nodes are also bridge nodes. The moment each cluster head apprehends its role in network, it broadcast a message through the network saying it role and it unique identity. The message is broadcasted by every node until it reaches a leaf node, a bridge node or cluster head. In case a leaf node receives two messages from different cluster head nodes, then it becomes a bridge node. The alternation of the leaf node role is occurred from the fact that its associated with two cluster heads.

3.2 Routing flag

After the network is formed and some requests have already disseminated through network, there are nodes which received considerably messages. Apparently, these messages were send from various nodes, a part of them is bridge nodes and some are cluster heads or from any other node. In case a received message is coming from a bridge node then it should be forwarded to nodes with high LC value. Otherwise if the message is coming from a cluster head it should be send towards nodes with high EC value.

Under those circumstances the custodian (node is in possession of the request), is not able to know how to disseminate the requests. With this in mind is added in the message data a one-bit flag which can take the values ‘U’ and ‘D’ in order to help the nodes figure out where to send each message. The ‘U’ value refers to messages going upwards to cluster heads and the ‘D’ value is for the messages that are forwarded downwards to bridge nodes.

First, all the messages initialize their flag as ‘U’ for the reason that originally the messages are forwarded to the cluster head of their team and afterwards to the rest of the network. In due time, each message changes this value, when it reaches nodes with specific roles. Specifically, cluster heads before sending a message change the flag value to ‘D’. Similarly, bridge and leaf nodes are changing the flag value into ‘U’ before forwarding a message. As a result of this, the intermediate nodes are taking advantage of the flag and efficiently disseminate each message to the accurate direction. To conclude, a bitwise flag is included to the message of each survivor in order to get routed across disperse locations effectively. This implementation helps to avoid sending a message in the direction that came from.

3.3 Sending technique

The sending technique differs between the nodes that have dissimilar roles in the network. As a result of this the dissemination policy is producing a desirable result. Initially the help-requests are forwarding upwards to the cluster heads. Following on are forwarded down to bridge nodes in order to end up at other clusters. Considering that some messages may be in different stages, at the same time some messages are

going to forwarded upwards and some downwards, is included a flag value in the message. The use of the flag is to define the stage of the message. Flags of the messages are altered from particular nodes.

3.3.1 Cluster head sending technique

Nodes that estimate their role as cluster head, are forwarding the messages from their database to the active neighboring nodes with the highest EC value. Before a cluster head forward a message, it initializes the flag of the message as 'D' which means downwards. The logic behind this action is that the message has reach the top of the cluster and then it has to be routed to other clusters. Consequently, the message should be transmitted downwards to a bridge node.

3.3.2 Leaf nodes sending technique

The leaf nodes should send the requests to the cluster head of their cluster. Thus, are sending their messages to their immediate neighbor which has the highest LC value and is active at the moment. In addition, the leaf nodes modify the flag of the message to 'U', meaning that the message is going upwards. Since, leaf nodes are the nodes with the lowest LC value, is reasonable that their message is going to be send up. So nodes with bigger LC value receive it and then forward it towards cluster heads.

3.3.3 Bridge nodes sending technique

Due to the fact bridge nodes connect two or more clusters, is intricate to realize where to forward a message. Is unattainable to discriminate which of his neighbors appertain in each cluster. As a consequence, bridge nodes broadcast their messages to all of their active neighbors except from the nodes that already received that message. In that case, the nodes that received the message can reroute it towards their cluster head. All clusters will receive that message with this strategy. Additionally, the flag of the message is taking the value 'U' to help the dissemination towards the cluster heads.

3.3.4 Intermediate nodes sending technique

A remarkable observation is that intermediate nodes are represent the paths between bridge nodes and cluster head. Accordingly, intermediate nodes forward request from up to down and reverse. Having this in mind, is wise to use the flag in the messages to comprehend where to disseminate each request. If the flag of a message is 'U' then the message is forwarded to the neighbor with the highest LC value. Because the flag gets this value from bridge and leaf nodes in order to lead the message to the cluster heads. Otherwise the message comes from cluster heads and is forwarded to the neighbor with the highest EC value to end up in bridge nodes.

3.4 EC values alternation

As we explained before a bridge node is possible to be related with more than two clusters. By the same token, in a cluster is probable to belong more than one bridge nodes. We need to take into consideration that each request is forwarded only once from a node except if a new candidate comes forward.
Thereafter a cluster head is going to forward the requests only to one of the bridge nodes that are member in it cluster. Knowing this, the calculation of EC values is recalculated after a time period with different Φ value. Since the Φ value is a random number between 3500 and 4500 add up to EC values of bridge nodes, after approximately a 10-minute period the Φ value is changed for each bridge node. By doing this the bridge nodes are changing values after a while, and the cluster heads are discovering new candidate to send the requests.
Assuming that there are two bridge nodes in a cluster and initially the Φ1 (the Φ value of the first bridge node) is higher than Φ2 (the Φ value of the second bridge node) the cluster head is going to forward the requests towards the first bridge node. The next time of the calculation is possible the Φ2 gets a higher value than Φ1, so the leader of the cluster is going to forward the messages to the second bridge node. As a result of this, the requests are disseminated to two different clusters.

On the condition that bridge nodes are receiving the requests, then the requests are forwarded to clusters that bridge nodes belong. So, the recalculation is done to ensure the dissemination of the requests to all the neighboring clusters.

3.5 Start mode and cycles

Is essential to mention the cause of the modes existence, which is the longevity of the network. If each node is in sleep mode for some time, it maintains the remaining battery of the device.
Initially, the nodes calculate a random value with the maximum of to be the cycle's duration. The value is representing the active cycle, scilicet the time period the node is going to be active for the cycle duration. For the rest of the cycle's duration the node is in sleep mode. In case the remaining battery of the device is reduce to the half or less, the active period time is lower than the half of the cycle value. In other words, if the active cycle that is randomly initialized is less than the half of the cycle value the most of the time the node is in sleep mode. Resulting in long durability of the device and significand battery saving.
The cycle value is in minutes and usually is initialized as one hour (60 minutes). Also, the value is fixed for all the devices. Additionally, the nodes mode at the beginning of the cycle, is randomly initialized. At the start of the cycle, part of the nodes may be in active mode and other in sleep mode.
The mode value and the cycle value of each node are reinitialized every 60 minutes, before the cycle start.

3.6 Already send or not value

When is time to send a message, the devices select the best candidate to send the specific message and if it already has it then the message is dismissed.
But sometimes the messages gets stack onto a node and never gets forwarder further. This problem incumbent upon to the fact that the candidate to receive the message is a node from the region that the message came from.
To confront this problem when a node receives the message, a one-bit value at the database of the device is initialized as 0. Before the node is going to send the message, if the value is zero, the candidates to send the message are the immediate neighbors of the node that never received the specific message. In case the value is 1 the forwarding operation is the same as before.
The purpose of this operation is to avoid the cycles between a number of nodes.

Chapter 4

Algorithm Evaluation

For the purpose of evaluating and finding the best possible algorithm, we have made multiple simulations of different scenarios. Firstly, an ideal scenario was created in order to see in detail the forwarding technique through the network, the consuming of the battery and other metrics which are going to be analyzed below. Afterwards, we made alternations with the intention to observe the changes of the network behavior. All the scenarios are representing a network of 400 nodes. The time period of the scenarios are 3600 minutes, in other words two and a half days. Furthermore, the positions are randomly initialized for each node inside a cycle of a 2000m diameter. In all the cases all the nodes send a rescue message. Each node send it message only once, except if a better candidate arises as the network changes.

The scenarios we are going to analyze are:

A. Ideal Scenario

B. The A scenario with a change in signaling operation
C. The B scenario without the one-bit operation to ensure the forward of all the requests
D. Real scenario, a more representative scenario of the reality

Is important to notice that for all the scenarios considered here, on average 11.5% of the nodes are cluster head, 68.5% are intermediate nodes and the remaining 20% leaf or bridge. The 20% is divided into 16.5% of bridge nodes and 3.5% of leaf nodes. Is noticeable that the intermediate nodes are by far more than the rest of the roles in network. Also the 82.704% of the initially leaf nodes, end up as bridge nodes, which is helpful for the dissemination policy of our algorithm.
To conclude, initially an ideal scenario was created in order to have the big picture and optimize the results it produced. Analogous to the observations we made of it results, we have made changes to see how the behavior of the routing technique is altered. To end a realistic scenario was created in order to check the results of the algorithm in its final use.
Afterwards, in order to see the reachability for the network with less rescue messages, we made some additional cases based on the first scenario. In this cases, from the network of 400 nodes, only a number of them have a rescue message to send. With that we aim to see the evolution of the network as the number of the nodes that have a rescue message increases.

4.1 Scenario A (Ideal)

The A scenario is the ideal one. The purpose of it creation is to fully understand how the algorithm is working, and if there are faults in the sequence of the dissemination policy.
We have named this scenario as ideal because the battery of all the devices is full and the devices are active for the entire duration of the simulation. The signaling technique is included in this scenario with some alternations from the description in chapter 3, the signal of the cluster head is forwarded by other cluster heads and the broadcasting does not stop there. Also in this scenario is included the routing flag whose use is explained in the sub-chapter 3.3. Additionally, all the nodes must send every message they have

received if the circumstances allow it. Last is included the operation of EC values alternation so the cluster heads forward their messages to all the bridge nodes of the cluster.
The metrics below are collected in order to evaluate the efficiency of the algorithm and find it weaknesses.

4.1.1 Reachable cluster heads

In the following table we can see the percentage of cluster heads that received the message of the nodes. In the best case scenario, the messages should be received from all the cluster head, because they are the nodes with the best probabilities of longevity. Consequently, when the rescuers discover the area of the network to help the survivors, cluster heads are the nodes whose database is going to be transmitted to them and help them locate the stranded survivors.

	Reachable cluster heads percentage				
Nodes	**All to cluster heads**	**Cluster Head to cluster heads**	**Intermediate to cluster heads**	**Leaf to cluster heads**	**Bridge to cluster heads**
%	92.22	89.4	92.69	94.53	91.2

Table 4.1 Reachable cluster heads percentage for scenario A

From the data of the table, is obvious that the percentage of cluster heads which received each message is high. Comparatively, the leaf nodes have higher percentage of the other roles and cluster heads have the lowest. The percentage of cluster heads that received the messages of all the nodes on average is 92.22%.

4.1.2 Fail to send

Under some circumstances, some requests may not be sent to other nodes. This is happening if there are not devices in it reachable range of the node to send it message.

As we can see from the table, the number of nodes in this category is zero for this scenario. This is due to the fact the average degree of connectivity in this network is six. Resulting in a very connected graph. There are parts of the network that are not connected to the main graph, but are smallest graphs of 4 to 5 nodes.

	Percentage of nodes that never send their message				
Nodes	**All**	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
%	0	0	0	0	0

Table 4.2 Send Failure for scenario A

4.1.3 Power evolution

For the simulation purposes, the battery was decreasing when the devices were sending or receiving requests. The battery consumption when the nodes are sending requests is twice the amount of consumption when receiving the requests

	Average remaining battery				
Nodes	**All**	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
Power	310.45	310.75	310.4	310.25	310.55

Table 4.3 Remaining battery on average for scenario A

Initially the average power of all the nodes is 1000. We assumed that the devices are all fully charged and the battery capacity is 1000. The battery of all the nodes was decreasing stably and with the same rate.

Finally, the remaining power of all the nodes on average is 310.45, is safe to say that the mean battery of all the nodes is dropped off by about 70%. The mean battery for all the groups of nodes, after the end of the simulation is pretty much the same.

4.1.4 Hops histogram

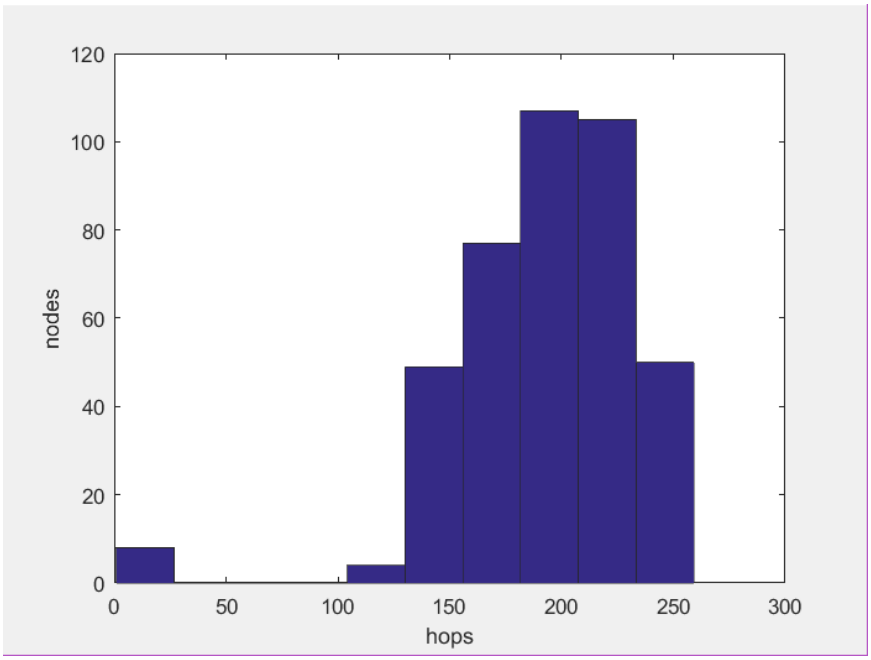

Image 4.1 hops histogram for scenario A

The significance of this graph is to observe the average number of hops the messages do in order to finally arrive in the receiver device. Is really important for our study to disseminate the requests deep through the network. Since the probability to receive the requests other cluster heads are better. If more cluster heads receive the request of a device then the probabilities to discover the survivor which the device belong to, are rises. Since the first aid responders can read it from more discoverable nodes.

As we can see generally the average hops away of the messages each node received is high and it fluctuates between 104.2 and 259.15 hops. Statistically, the graph is clarifying in the table:

	Majority of nodes	**Minority of nodes**	**Most hops**	**Fewer hops**
Nodes	107	4	50	8
% nodes	26.75%	1%	12.5%	2%
Hops	183 - 207	104.2 – 130	233.32 - 259.15	1 - 26.75

table 4.4 analyzing hops histogram for scenario A

4.1.5 Received messages

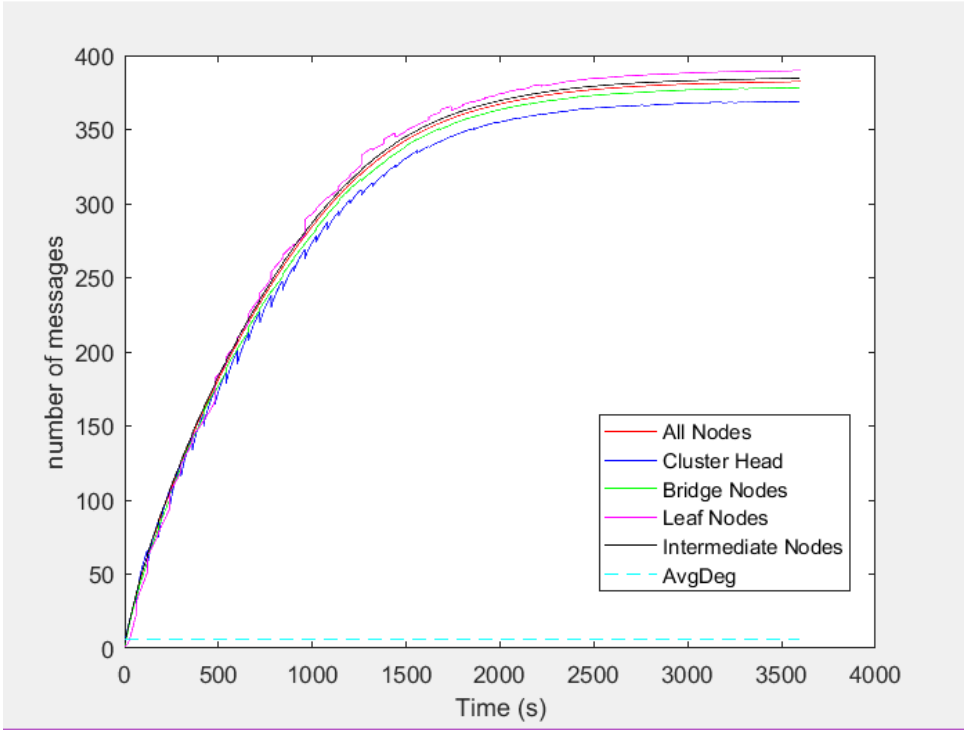

Image 4.2 Number of received messages for scenario A

The graph demonstrates the average number of the received messages on four different roles of nodes compared with time. Specifically, the number of rescued messages that the nodes received. In this scenario all the nodes have a rescue message to send. The more messages the nodes have, when the rescuers discover them, the more stranded survivors are going to locate.

Throughout the period of the simulation the number of messages received from each node is increasing. At the end the average number of received messages for the leaf nodes is 389.72. Comparatively with the other groups of nodes is the highest value. The cluster heads have on average 368.65 which is the lowest. The intermediate nodes and the bridge nodes have 384.54 and 377.87 respectively. For all the nodes of the network eventually the average of their received messages is 382.3.

4.1.6 Sent messages versus time

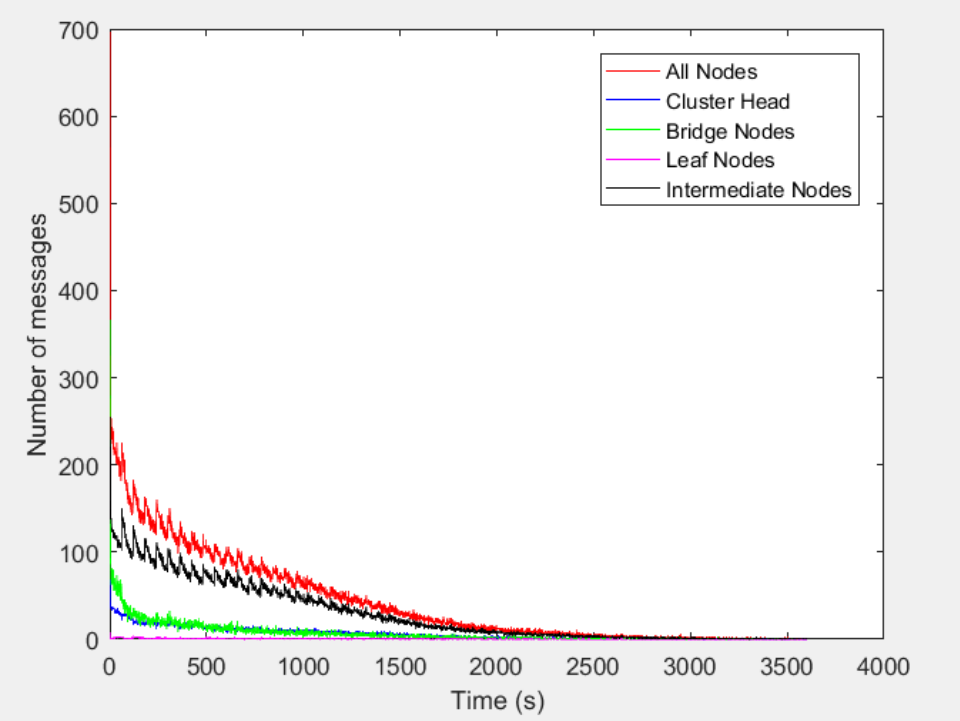

Image 4.3 number of messages send each iteration for scenario A

The graph illustrates the number of messages send in each minute. Is important to realize as the time passes and when a significance number of messages is sent, if the forwarding of the messages evolves or remains stable. If the number of sent messages changes analogous with time, that means the messages are flowing in the network and there is an evolution of the message forwarding.

Initially the send messages are numerous. As we approaching the end of the simulation the send messages number is getting smaller. This is happening because as the end of the simulation is coming the biggest part of the dissemination is already done, the most of the nodes have already send all their received messages. Eventually all the messages have sent throughout the network, so there is nowhere else to get forwarded.

The total messages send for the whole scenario are represented in the following table:

	Number of send messages for the duration of 3600 minutes				
Nodes	**All**	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
Messages No	152545.4	22473.8	104844	1430.2	23797.4

Table 4.5 messages send for scenario A

	Percentage of the total number of messages send			
Nodes	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
%	14.73	68.72	0.93	15.60

Table 4.6 percentage of the total number of messages send for scenario A

The greater amount of messages is send from the intermediate nodes, which is logical since all the messages that are send from the cluster heads to the bridge nodes and reverse have to pass from the intermediate nodes. The less messages have been sent from the leaf nodes.

4.1.7 Cluster head path for scenario A

In the image above we can see the path of the node 362. The role of the node 362 in the network is cluster head. The pink lines in the picture show the route of the message, if a line that links two nodes is pink it means that the message of the node 362 was sent from the one of them to the other.
As it seems from the image, the message of the node 362 is sent throughout the whole network and is received from the majority of the nodes.

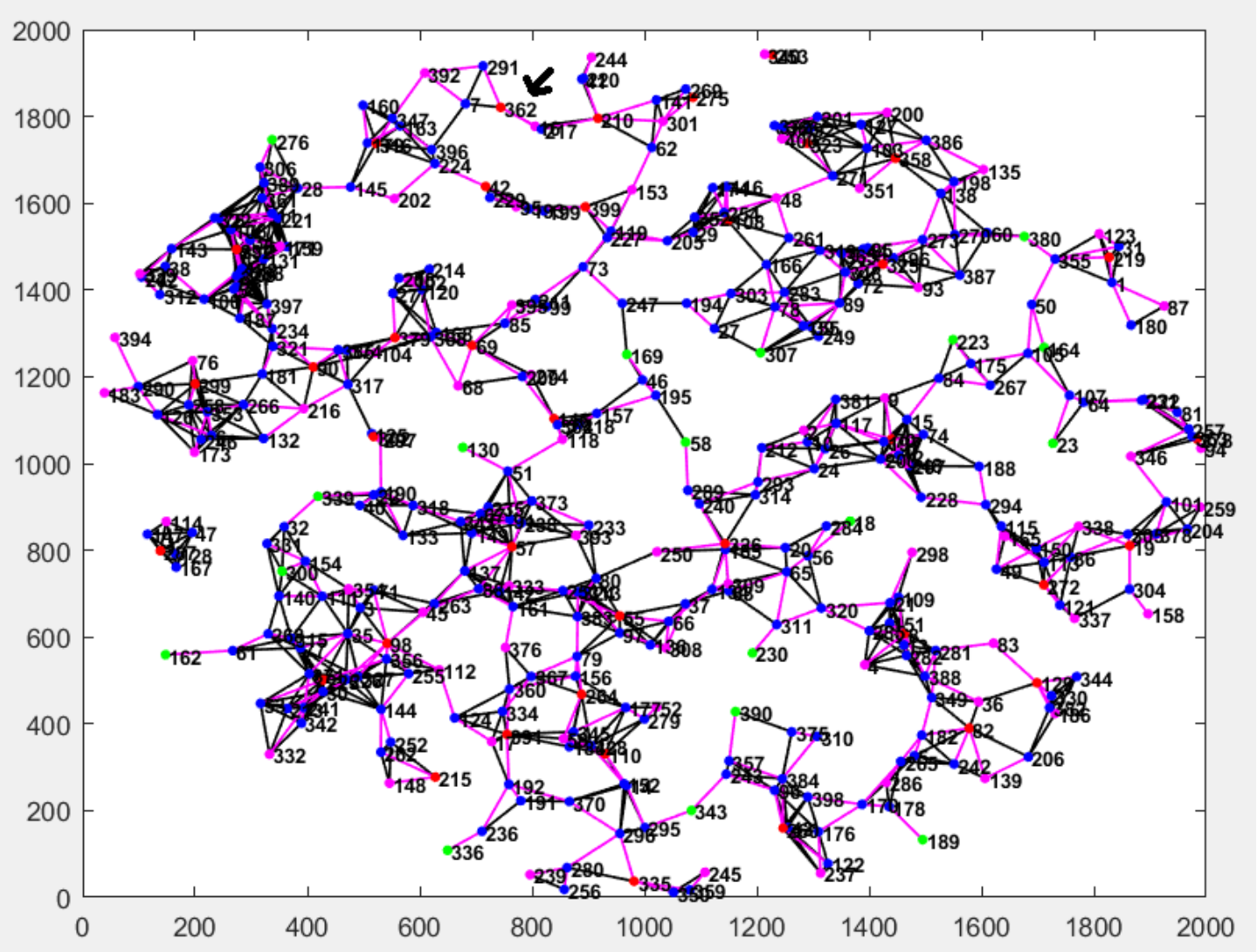

Image 4.4 cluster head path for scenario A

4.2 Scenario B

This scenario is made to evaluate the sufficiency of the algorithm when all the devices has full residual battery and are all active the whole duration of the simulation. It is an exact duplicate of scenario A, specifically the routing flag, the signaling method and the EC values alternation is included. However, in this scenario the signaling technique is implemented exactly as is explained in chapter 3. By omitting the cluster heads in signaling technique we seek to decrease the overhead of the signals. The signals overhead is dropped 18.44 per cent compared with the previous scenario.
Below are analyzed the same metrics with scenario A to check if the alternation we made drop the performance of the algorithm

4.2.1 Reachable cluster heads

The table give particulars about the proportions of the cluster heads that received the message of the nodes. The ideal is the messages end up to all the cluster heads, since are the nodes with the most residual battery and they can stay “alive” for a longer period of time.

	Reachable cluster heads percentage				
Nodes	**All to cluster heads**	**Cluster Head to cluster heads**	**Intermediate to cluster heads**	**Leaf to cluster heads**	**Bridge to cluster heads**
%	89.93%	83.37%	90.97%	91.5%	89.213%

Table 4.7 Reachable cluster heads percentage for scenario B

From the data of the table we conclude that the leaf nodes have the higher percentage and the cluster heads have the lower. Furthermore, the proportion of the cluster heads that received the messages of all the nodes is 89.93 per cent, which is lower than the previous scenario, but the difference is very small. The exact difference between the percentages of the two scenarios is 2.29. However, the preferable result which is the 100% of cluster heads to receive the messages is still close to the percentages of this scenario.

4.2.2 Fail to send

	Percentage of nodes that never send their message				
Nodes	**All**	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
%	0.25	0.25	0	0	0

Table 4.8 Send Failure for scenario B

At the table is represented the percentage of nodes whose messages did not have the opportunity to be sent.

As it seems from the table a small portion of nodes and in particular of cluster heads, never sent it message towards others nodes. This proportion is emerged from some nodes which do not have neighboring nodes. Scilicet there are no other nodes in their reachable range and this causing them to stay undetectable.

4.2.3 Power evolution

Is important to point out that when the simulation starts the average battery of all the nodes is 1000 since the higher value the nodes power can take is 1000 and we assumed that all the nodes are fully charged at the beginning. For the duration of the simulation the battery of the devices is decreased linearly.

	Average remaining battery				
Nodes	**All**	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
Power	310.45	311.07	310.3265	310.3625	310.631

Table 4.9 Remaining battery on average for scenario B

From the table data is obvious that the remaining power of the groups of nodes is about the same. The average remaining battery capacity of the cluster heads is higher from the other groups of nodes. To conclude, is important to mention that the average power of all the nodes is decreased by 68.955% till the end of the simulation.

4.2.4 Hops histogram

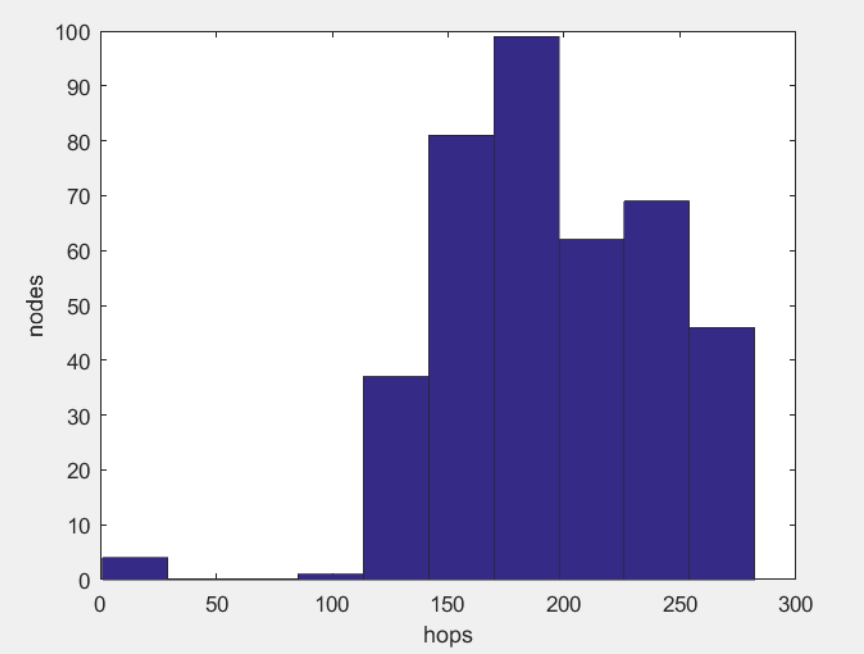

Image 4.5 hops histogram for scenario B

The bar chart represents the average hops the received messages did to end up to the nodes. The probabilities of the messages to receive from more cluster heads is analogous to the hops the messages will do. As a result, the potentiality to be found is higher.

Basically, the average of the messages that are received from the most nodes is high, approximately between 85 – 282. The highlights of the graph are shown in the table below.

	Majority of nodes	**Minority of nodes**	**Most hops**	**Fewer hops**
Nodes	99	1	46	4
% nodes	24.75%	0.25%	11.5%	1%
Hops	169.7 – 197.82	85.347 – 113.462	254.04 – 282.155	1 – 29.12

Table 4.10 analyzing hops histogram for scenario B

4.2.5 Received messages

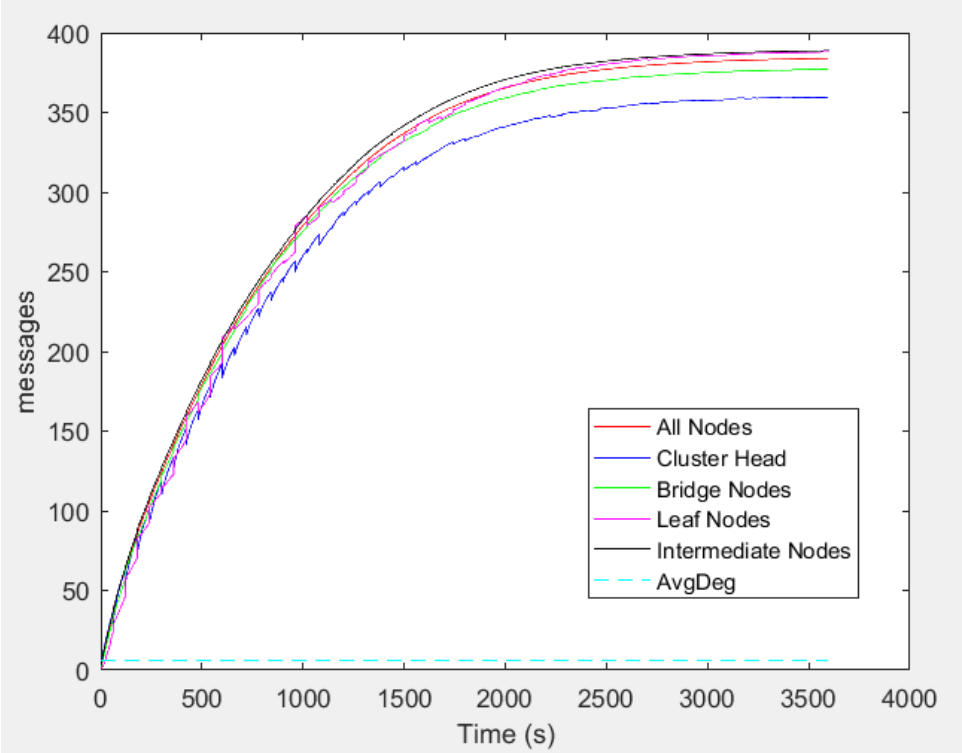

Image 4.6 Number of received messages for scenario B

The graph shows the number of received messages on average for the four groups of nodes. The more messages the cluster head that is discovered from the rescuers has received, the more stranded survivors are going to be found.

The intermediate nodes have received on average more messages at the end relatively with the other group of nodes. Which are 388.833 messages on average. The lowest value corresponds to the group of cluster heads and is 359.43. Eventually, the bridge nodes have on average received 377.125 messages and the leaf nodes 387.815 messages.

Averagely, all the nodes have received 383.9 messages, which is a really good achievement considering that all the messages are 400 and the average connectivity is 6.

4.2.6 Sent messages versus time

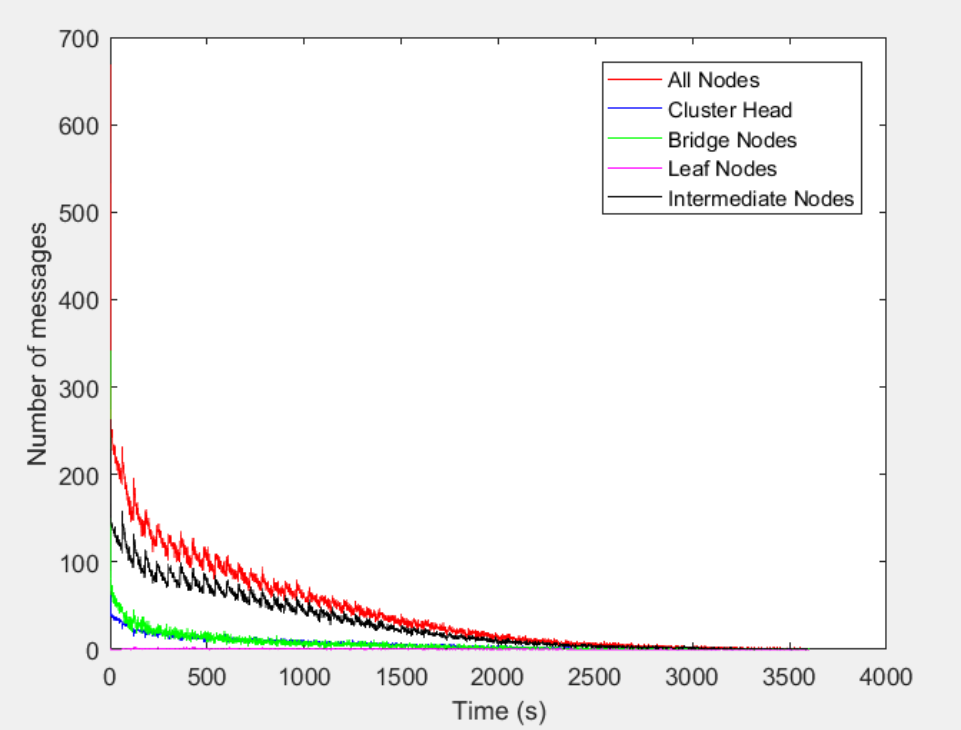

Image 4.7 number of messages send each iteration for scenario B

The graph exhibit the number of messages send in each minutes in the whole network. With the time passing the number of messages send is diminishing. The intermediate nodes are sending more messages in each iteration than the other group of nodes. The first minutes the number of the messages send is extremely high in contrast with the rest of the time.

The number of messages send in the whole simulation is represented in the table. Is important to note that the intermediate nodes are sending the majority of the messages and the leaf nodes the less messages.

	Number of send messages for the duration of 3600 minutes				
Nodes	**All**	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
Messages No	153172.4	21973	106263.4	1249	23687

Table 4.11 messages send for scenario B

	Percentage of the total number of messages send			
Nodes	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
%	14.34	69.375	0.815	15.46

Table 4.12 percentage of the total number of messages send for scenario B

4.2.7 Bridge node path for scenario B

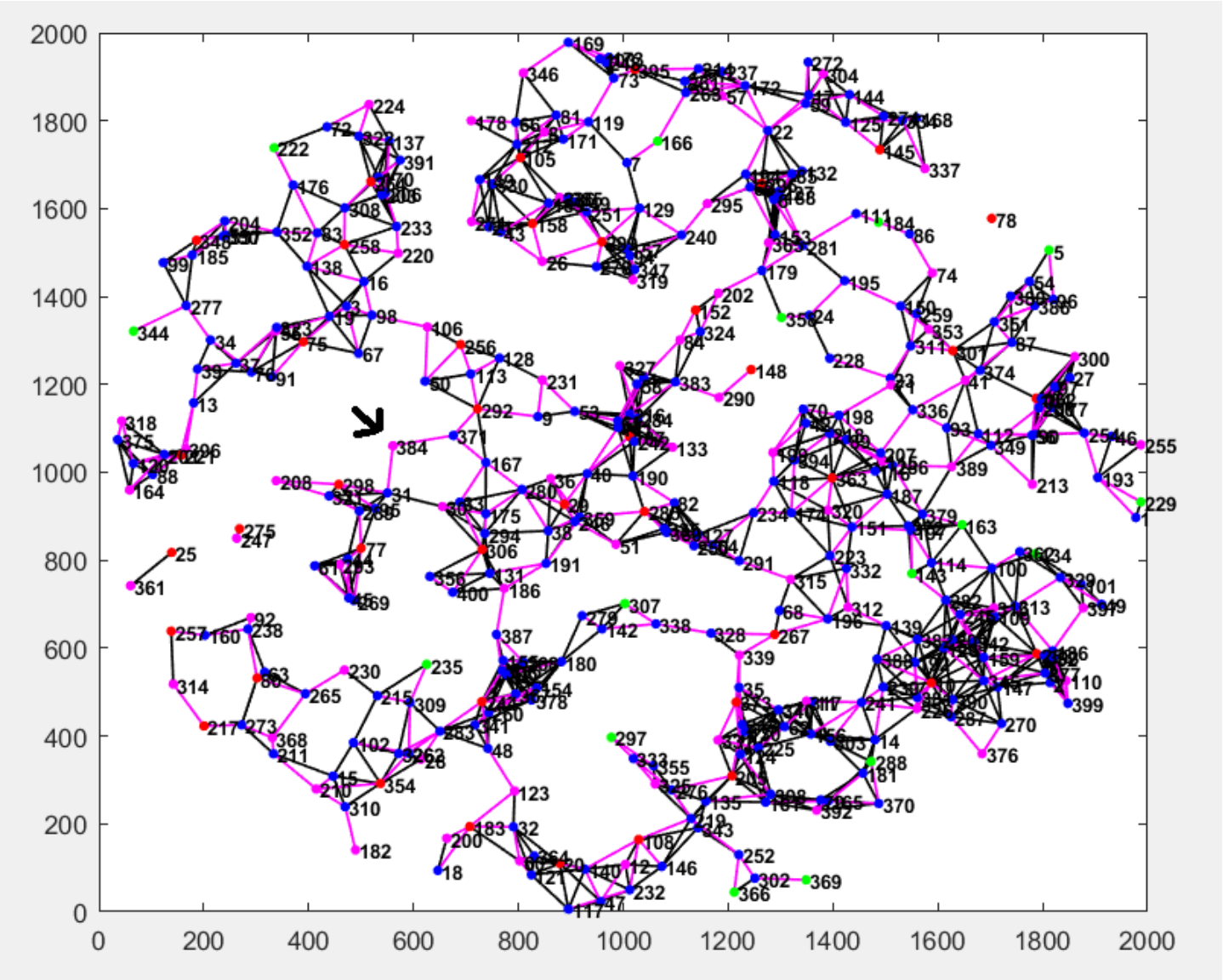

Image 4.8 bridge node path for scenario B

In the image above is represented the path of the message from the node 384. The role of the node 384 in the network is bridge.

The pink lines are showing the path of the message in the graph. It clearly seems that the message was received from many nodes in the graph and from the most cluster heads.

4.3 Scenario C

The third scenario uses the same techniques as the second one, the battery of all the devices is full at the beginning of the simulation and the devices are active until the end, the sleep mode is not used. Moreover, the signaling technique, the routing flag and the EC value alternation is also included in this scenario.
The change made for this scenario is in the "already send or not operation". We assumed that when a node receives a message is mandatory to forward it even though the appropriate receiver already has it. In this scenario, the messages are not forwarded if the best candidate already has it even if the specific node never send it.

4.3.1 Reachable cluster heads

Below we show in detail the average percentage of cluster heads that received the messages for all the groups of nodes. Is the more important metric of the simulations, since the requested of the algorithm is all the cluster heads to receive the message of each node. By accomplishing a high percentage of cluster heads, the probabilities of stranded survivors to get rescued are more.

	Reachable cluster heads percentage				
Nodes	**All to cluster heads**	**Cluster Head to cluster heads**	**Intermediate to cluster heads**	**Leaf to cluster heads**	**Bridge to cluster heads**
%	92.56	89.3794	93.0673	94.3202	91.6703

Table 4.13 Reachable cluster heads percentage for scenario C

As we can see from the table, the leaf nodes and the cluster heads have the higher and the lowest percentage respectively.
The proportion of the cluster heads that received the messages of all the nodes is 92.56%

4.3.2 Fail to send

As explained before sometimes the graph is not fully connected, resulting in some nodes not to be connected with others. In this scenario as it seems in the table below, the graph created from the devices is fully connected. This upshot is supported by the metric of the average connectivity. The average connectivity of the graph created for this scenario is 6. So all the nodes send their message at least once.

	Percentage of nodes that never send their message				
Nodes	**All**	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
%	0	0	0	0	0

Table 4.14 Send Failure for scenario C

4.3.3 Power evolution

During the processes of receiving and sending message the battery of the devices is decreased. The battery of the nodes is decreased with stable rhythm. The following table is representing the remaining battery of the nodes at the end of the simulation.

	Average remaining battery				
Nodes	**All**	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
Power	310,4	310,75	310,35	310,25	310,6

Table 4.15 Remaining battery on average for scenario C

At the start of the simulation the average battery of all the nodes was 1000 since the devices were fully charged, at the end the battery was reduced by 68,96%. For all the roles the power at the end of the simulation is approximately the same.

4.3.4 Hops histogram

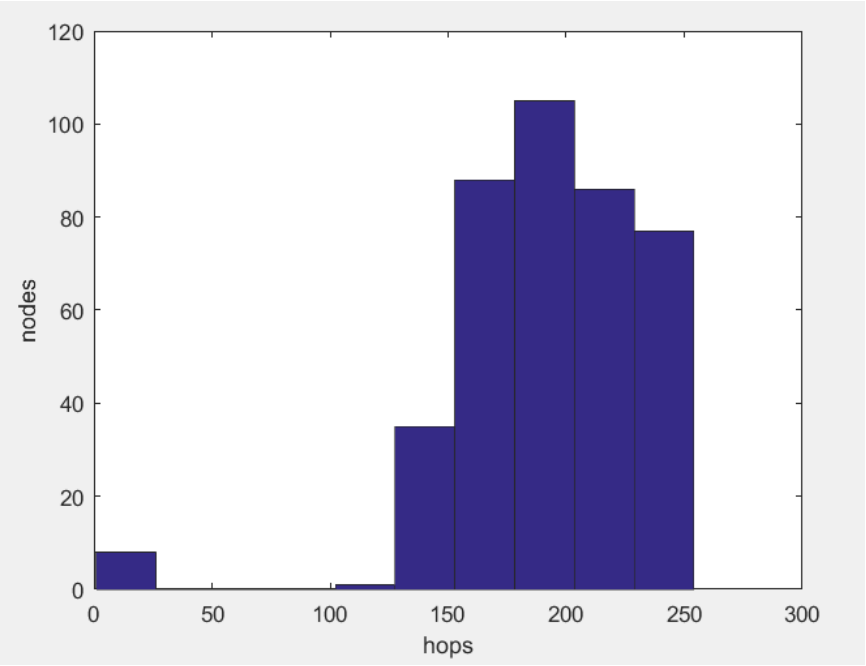

Image 4.9 hops histogram for scenario C

The chart gives information about the average hops the requests did in order to reach a specific node. A histogram is created for all the nodes.
The significance of the hops away is to grow the probabilities of localization by the rescuers.

	Majority of nodes	**Minority of nodes**	**Most hops**	**Fewer hops**
Nodes	105	1	77	8
% nodes	26.25%	0.25%	19.25%	2%
Hops	178.163 – 203.472	102.236 – 127.545	228.781 – 254.09	1 – 26.309

table 4.16 analyzing hops histogram for scenario C

According to the table the nodes have received messages from relatively many hops away. The proportion of nodes that received messages with fewer hops away is much smaller than the percentage of nodes with the most hops away.

4.3.5 Received messages

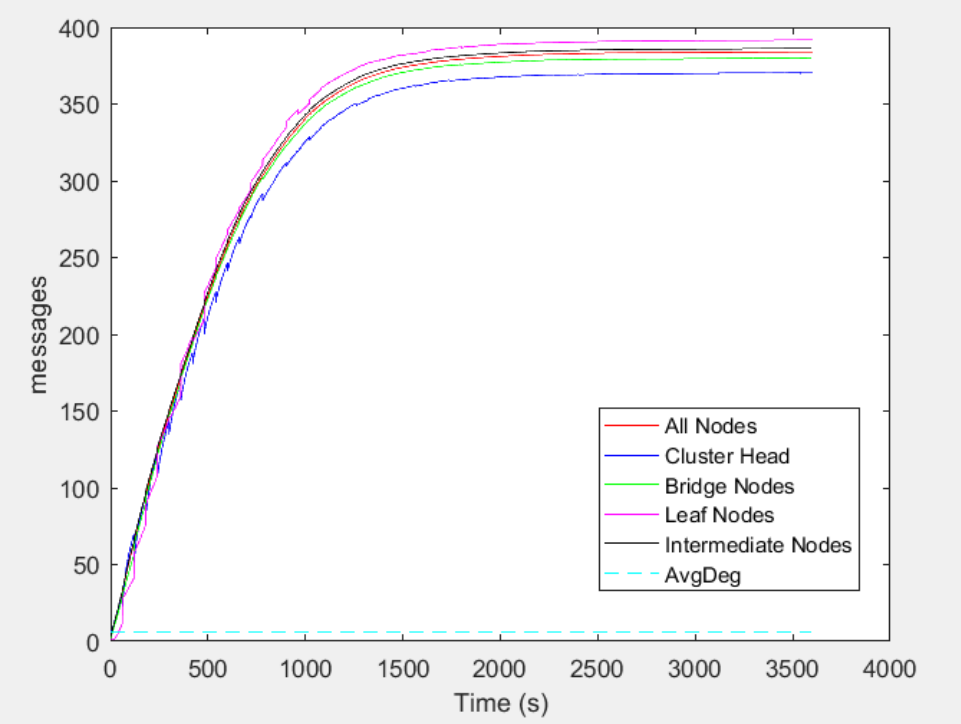

Image 4.10 Number of received messages for scenario C

The line graph clearly shows the number of messages the nodes have received for a period of time. The number goes up as the time passes.

Eventually the all the nodes have on average received 383.975 messages. The team of nodes with the most messages is the leaf nodes, who on average they have received 391.677 messages. Also the group of cluster heads have received 370.245 requests and is the group of nodes with the less messages. Furthermore, at the end the bridge nodes have received 379.969 messages and the intermediate nodes 386.2.

4.3.6 Sent messages versus time

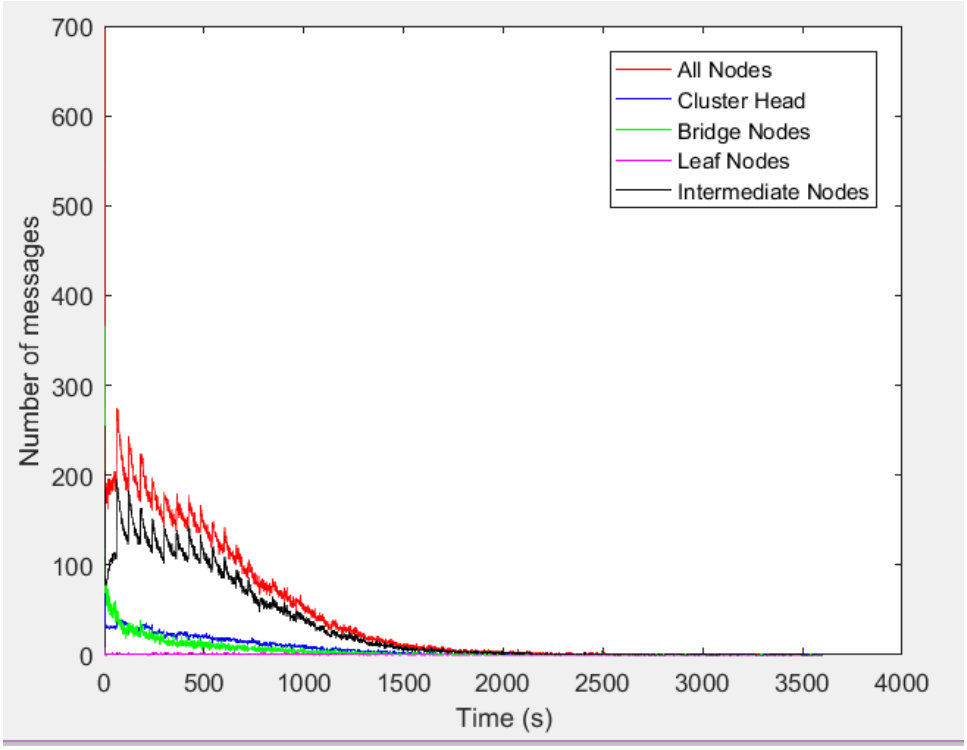

Image 4.11 number of messages send each iteration for scenario C

The graph is about the number of send messages on average, for each minute of the simulation. The y-axis shows the number of messages send and the x-axis the time. At the first minute of the simulation is send the largest number of messages. The number decreases as the simulation is getting closer to the end. For some iterations at the end of the simulation, the devices may not send messages. The number of messages that are sent is high since the number of rescue messages in the whole network are 400.

	Number of send messages for the duration of 3600 minutes				
Nodes	**All**	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
Messages No	153209.6	23831.6	110591.4	1322.4	17464.2

Table 4.17 messages send for scenario C

The largest number of messages is send from the intermediate nodes this result is

logical, since in the simulation we have heaps of intermediate nodes. Additionally, the path between the cluster heads and the bridge nodes is made up with intermediate nodes. Basically all the messages pass by this group of nodes, resulting in big number of sent messages. In contrast with the leaf nodes, who send the less messages comparatively with the other groups.

	Percentage of the total number of messages send			
Nodes	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
%	15.55	72.18	0.86	11.39

Table 4.18 percentage of the total number of messages send for scenario C

4.3.7 Leaf node path for scenario C

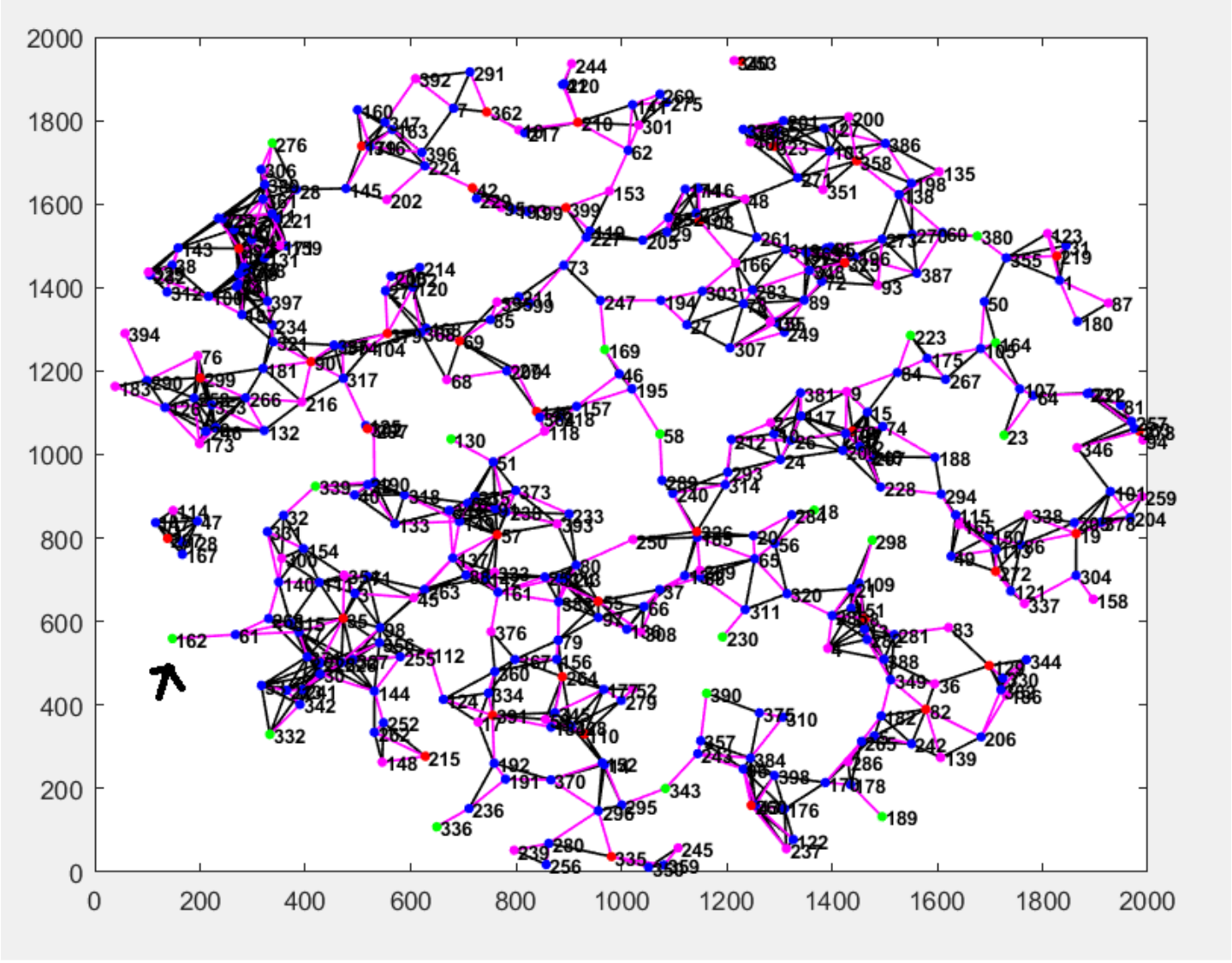

Image 4.12 leaf node path for scenario C

The image is a representative path of the message of the node 162. The node 162 in this scenario is leaf.
If a pink line links two nodes, that means the message of the node 162 is send from the one node to the other. From the image we can see that the message has travelled through the whole network.

4.4 Scenario D

The scenario was created to control the efficiency of the algorithm in real-life circumstances. For the scenario purposes the battery of the devices is a randomly initialized number between 250 and 1000. Also, the active cycle of the devices takes a random value before each cycle start. For the rest duration of the cycle the devices are inactive. Furthermore, the start mode of each node is randomly initialized. Scilicet if a node is going to be in active or sleep mode at the beginning of the cycle.
As is mentioned in the other scenarios, the signaling technique for discovering the bridge nodes is included. Additionally, the routing flag and the EC values alternation is used in this scenario. All the nodes of the created network must forward each message they received at least one time, if is achievable.

4.4.1 Reachable cluster heads

The quantity of cluster heads that received the message of a survivor is considered as the most important metric to evaluate the performance of the algorithm. The preferable is the message to be received from the full proportion of cluster heads.
In the table below are represented the percentages of the cluster heads that received the messages of each group of nodes.

	Reachable cluster heads percentage				
Nodes	**All to cluster heads**	**Cluster Head to cluster heads**	**Intermediate to cluster heads**	**Leaf to cluster heads**	**Bridge to cluster heads**
%	66.47	62.20	67.843	55.452	66.196

Table 4.19 Reachable cluster heads percentage for scenario D

So we can see that the leaf nodes have the lowest percentage and the intermediate nodes have the highest. Generally, the proportion are approximately at 60 per cent. The average cluster heads that received the messages of all the nodes is 66.47%.

4.4.2 Fail to send

Because of the random initialization of the nodes location, some nodes may be localized in a position with zero devices in their range. In this case, the nodes cannot send their message through the network, since no other device is reachable.
Even though the average connectivity of the graph created is 5.83, in the table it seems that some of the nodes, never send their messages. The number of this devices is really small and is negligible.

	Percentage of nodes that never send their message				
Nodes	**All**	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
%	0.75	0.75	0	0	0

Table 4.20 Send Failure for scenario D

4.4.3 Power evolution

For the purposes of this scenario the initial battery is not full for all the nodes.
Moreover, the sleep cycle of the nodes is included, so we expect the percentage of the device loss resources to be less than the previous scenarios. In the following table is shown the initial power of the roles and the remaining power after the simulation is ended.

	Average battery				
Nodes	**All**	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
Initial Power	584.8	584.8	584.8	584.8	584.8
Remaining power	140	145.8	139.9	135.75	133.7

Table 4.21 Battery on average for scenario D

The battery of the nodes decreases linearly. The remaining battery of all the groups of nodes at the end it pretty much the same. The power of all the nodes for the whole duration of the simulation is decreased by 23.94%.

4.4.4 Hops histogram

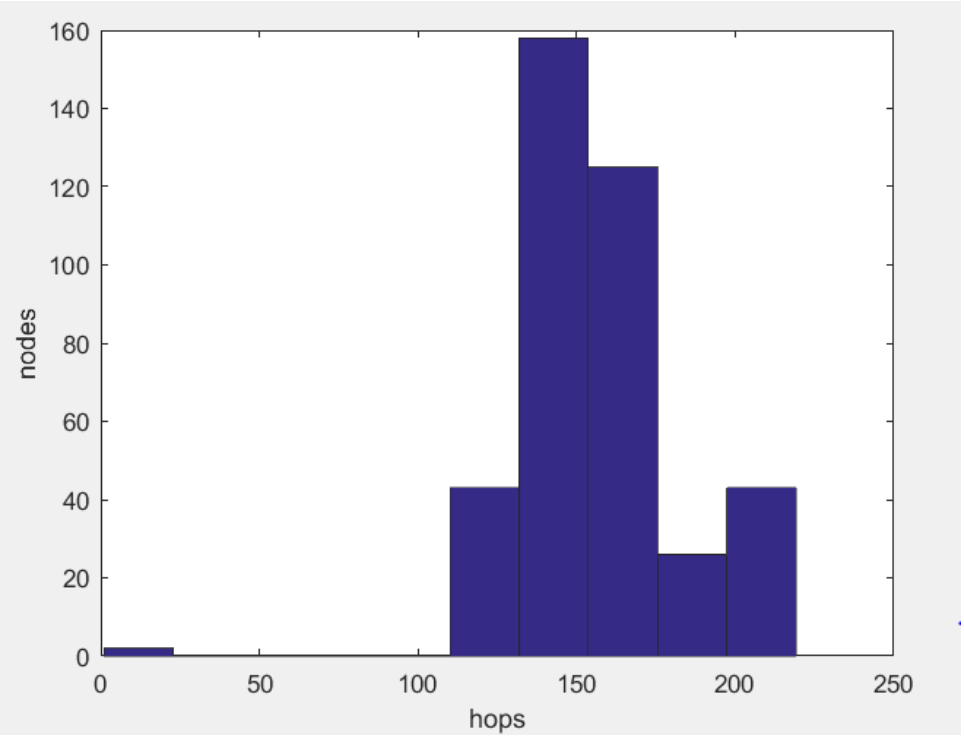

Image 4.13 hops histogram for scenario D

Initially is calculated the average hops away of the known messages for each node independently. After is created the histogram above representing the number of nodes that have the specific hops away number. In the table below we can see the graph highlights. The majority of nodes have messages from approximately 142.8 hops away. Scilicet the messages are received on average from 142-143 nodes before the node received it.

	Majority of nodes	**Minority of nodes**	**Most hops**	**Fewer hops**
Nodes	158	2	43	2
% nodes	39.5	0.5	10.75	0.5
Hops	131.9 – 153.715	1 – 22.815	197.35 – 219.16	1 – 22.815

table 4.22 Analyzing hops histogram for scenario D

4.4.5 Received messages

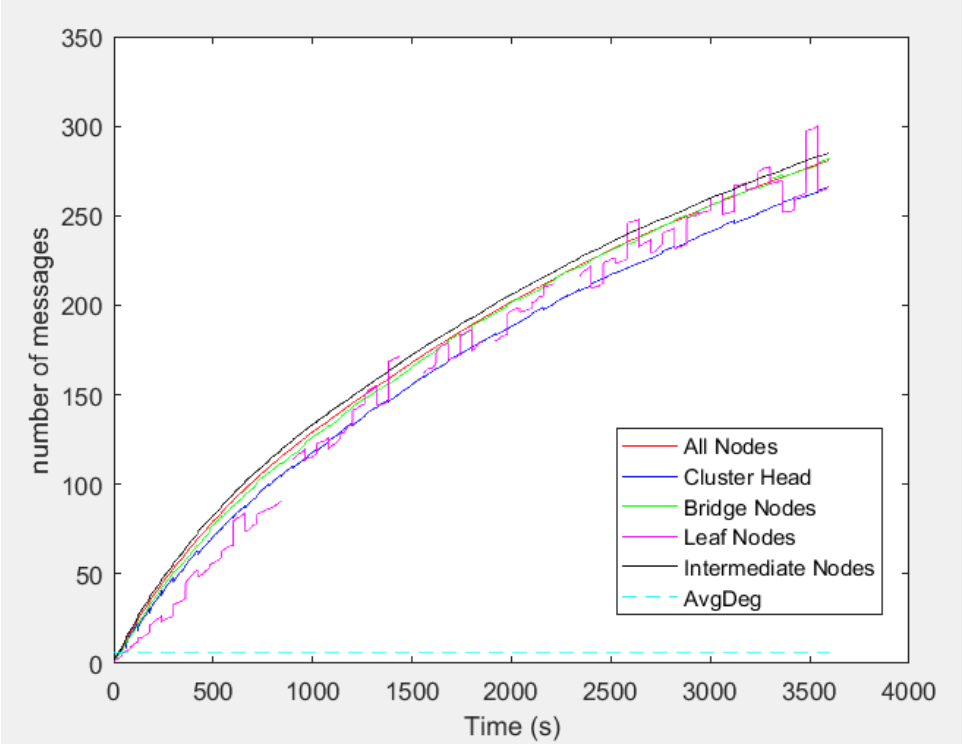

Image 4.14 Number of received messages for scenario D

The graph demonstrates the averagely number of received messages for all the group of nodes relative with time. The number grows up as the time is approaching the end of the simulation. When the simulation reaches the end the number of collected messages on average for the intermediate nodes is the highest with the value of 285.206 requests, in addition the lowest number of received messages is of the cluster heads with 266.293 messages. The collected requests of the cluster heads are really close with the leaf nodes, who received 266.85 requests. Furthermore, the bridge nodes have eventually received on average 282.065 messages. To conclude at the end of the simulation the mean number of received message for all the nodes is 281.116.

4.4.6 Sent messages versus time

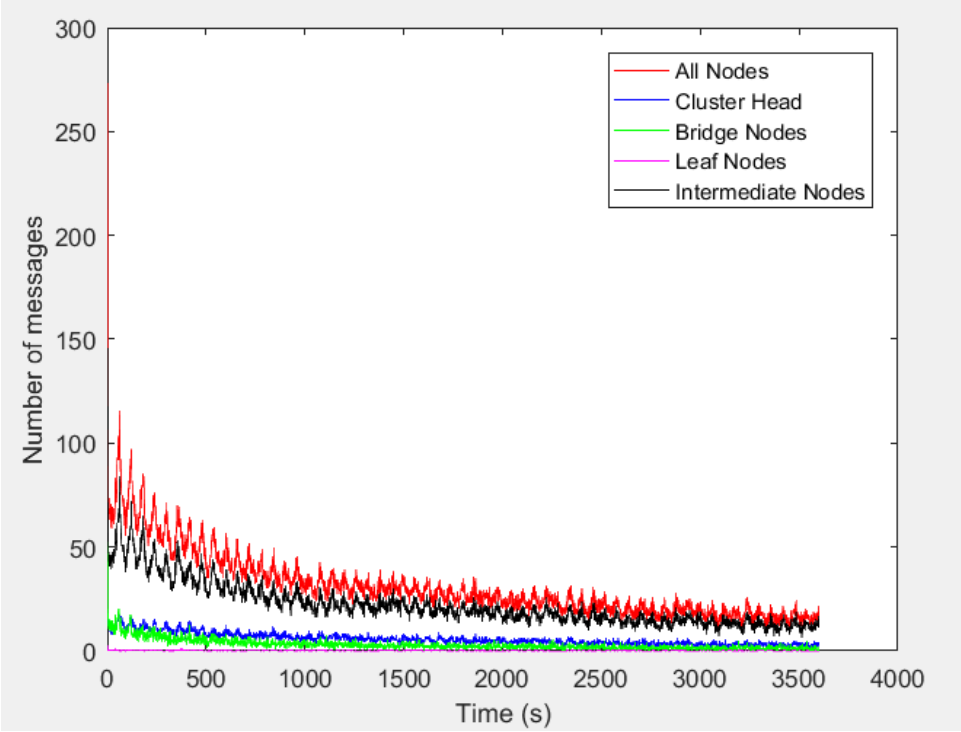

Image 4.15 number of messages send each iteration for scenario D

The graph above illustrates the number of messages send during the simulation. At the start of the simulation the send messages are extremely high in contradiction with the send messages at the end. Is obvious that the number of the sending messages is decreased versus time.

The number of the send messages for the whole duration of the simulation is represented at the table.

	Number of send messages for the duration of 3600 minutes				
Nodes	**All**	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
Messages No	112069	20641.6	80013.4	306	11108

Table 4.23 messages send for scenario D

The number of messages that are send by leaf nodes is the smallest and the number of messages send by the intermediate nodes is the largest. Comparatively with the other roles the difference is huge both for the leaf nodes and the intermediate nodes.

	Percentage of the total number of messages send			
Nodes	**Cluster Head**	**Intermediate**	**Leaf**	**Bridge**
%	18.418	71.396	0.273	9.91

Table 4.24 percentage of the total number of messages send for scenario D

4.4.7 Intermediate node path for scenario D

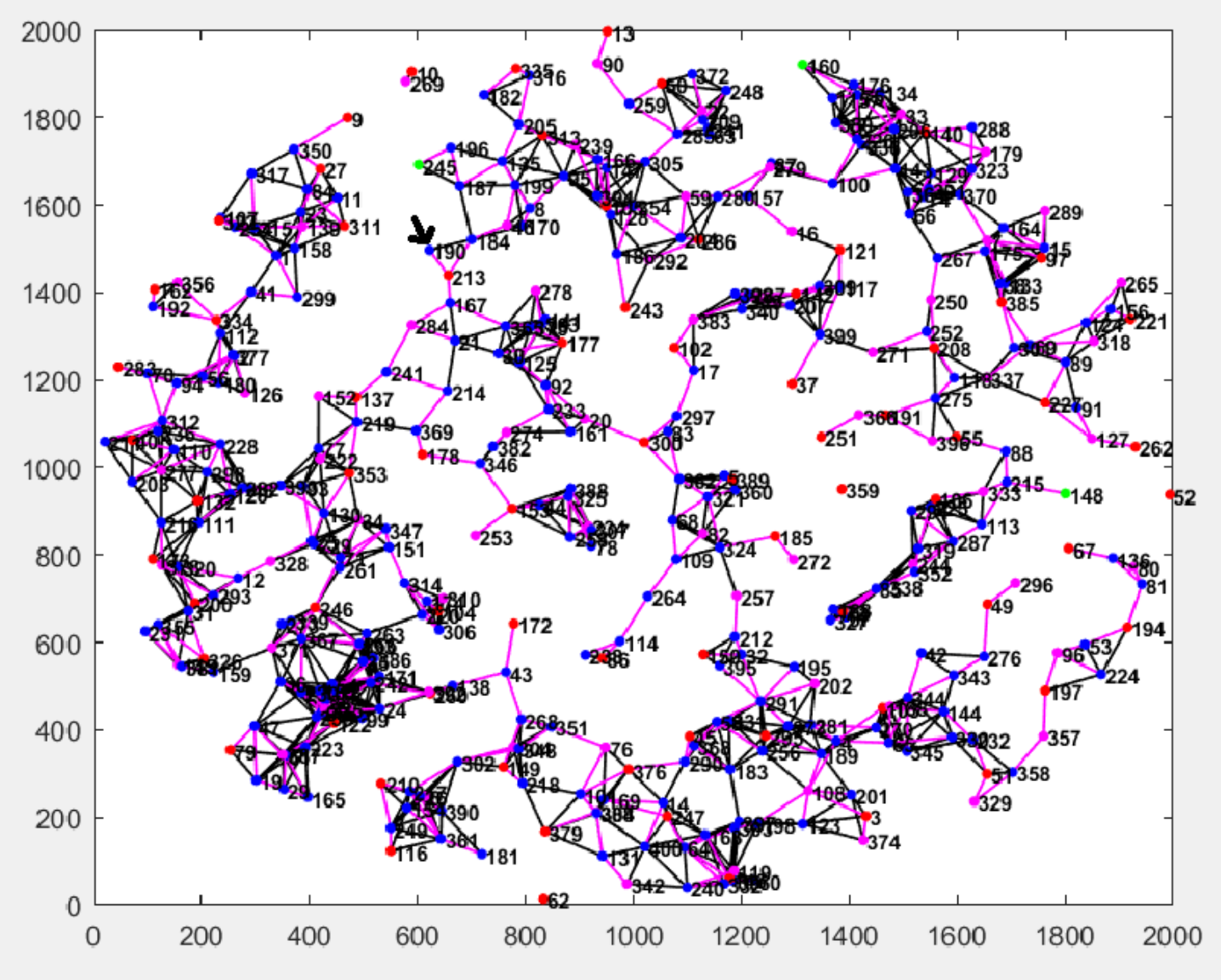

Image 4.16 Intermediate node path for scenario D

In this image is showing the route of the message of the node 190 which is an intermediate node.

From the pink lines is created a route, the message of the node 190 has passed from this path. The message has reached nodes from the entire network.

4.5 Summary of scenarios

The evaluation of the algorithm would not have been complete without compare data for the scenarios discussed above. The results below are represented for all the nodes, and the whole duration of the simulation.

4.5.1 Proportion of power decrement

	Power decrement			
Scenario	**A**	**B**	**C**	**D**
%	68.955	68.955	68.96	23.94

Table 4.25 power decrement for all scenarios

The table demonstrates the percentage of power decrement on average for all the nodes from the start of the simulation until the end. According to the table, the power decrement of the first two scenarios is exactly the same. Additionally, the third scenario has pretty much the same results with them. On the other hand, the battery reduce of the last scenario is significant smaller. Obviously the cycles (sleep and active mode) which are included in the last scenario in order to save power have positive results.

4.5.2 Reachable cluster heads of all nodes

	Percentage of cluster heads that received all messages			
Scenario	**A**	**B**	**C**	**D**
%	92.22	89.93	92.56	66.47

Table 4.26 reachable cluster heads for all scenarios

The table shows the percentage of cluster heads that received the messages from all the nodes for the four scenarios.
The proportion of the C scenario is higher than the others, but is really close with the one of the first scenario. So is safe to say that the change we made for the third scenario does not make a difference for the proportion.

The second scenario value is also close with the others two, but a bit lower, in difference with the percentage of the fourth one, which is much lower.

4.5.3 Sent messages

	Send messages for the whole duration			
Scenario	**A**	**B**	**C**	**D**
Messages number	152545.4	153172.4	153209.6	112069

Table 4.27 Sent messages number for all scenarios

The table illustrates the number of send messages from all the devices, for the total duration of the scenarios.
The send messages of the three first scenarios are approximately at the same level. Furthermore, at the last scenario have been send at least 40000 less messages from the others.

4.5.4 Received messages

	Received messages for the whole duration			
Scenario	**A**	**B**	**C**	**D**
Messages number	382.3	383.9	383.975	281.116

Table 4.28 Received messages number for all scenarios

In the table is demonstrated the number of received messages the devices had at the end of the simulation on average for all the nodes.
The data of the table indicates that the number of received messages on average of the second and third scenario are extremely close and the highest in compare with the other scenarios.
The received messages value of the first scenario is also around the same with the values of the two scenarios, but the number of messages for the fourth scenario is much lower.

4.5.5 Hops on average

	Average hops away for all the nodes			
Scenario	**A**	**B**	**C**	**D**
hops	191.8	194.8	192.1	157.1

Table 4.29 Hops away on average for all scenarios

In order to create the above table, firstly is calculated the number of hops-away that are made for each message to end up to a node for all the messages that the node has received. This value is calculated for all the nodes. Secondly, the average of this numbers is saved for each node. Last, the average for all the nodes is calculated and represented in the table. These calculations are done after the end of the simulation, when all the messages are already send.

It can be clearly seen that once again the result of the last scenario is lower than the others. From the other three scenarios the B has the highest value, and the two scenarios left have pretty much the same value.

4.6 Evolution of the network relative to time and the messages number

4.6.1 Snapshots for the evolution of one message

In a network of 400 nodes, is important for us to observe how much time is needed from a message to accomplish the goals we have set. Snapshots in different times of the simulation are included below to present the evolution of one message in the network, if is the only one sent in the graph.

From the images we can see that as the time passes, the message is sent towards more nodes. In the first image, which is a snapshot after 20 minutes of the network activity, the message is sent to the area near the node that the message belongs to. Afterwards, the message is sent farther from the starting point.

The dissemination of the message is ended at the 80 minutes, since after it is not sent further in the network. This is logical since the message after the period of the 80 minutes has travelled throughout the whole network.

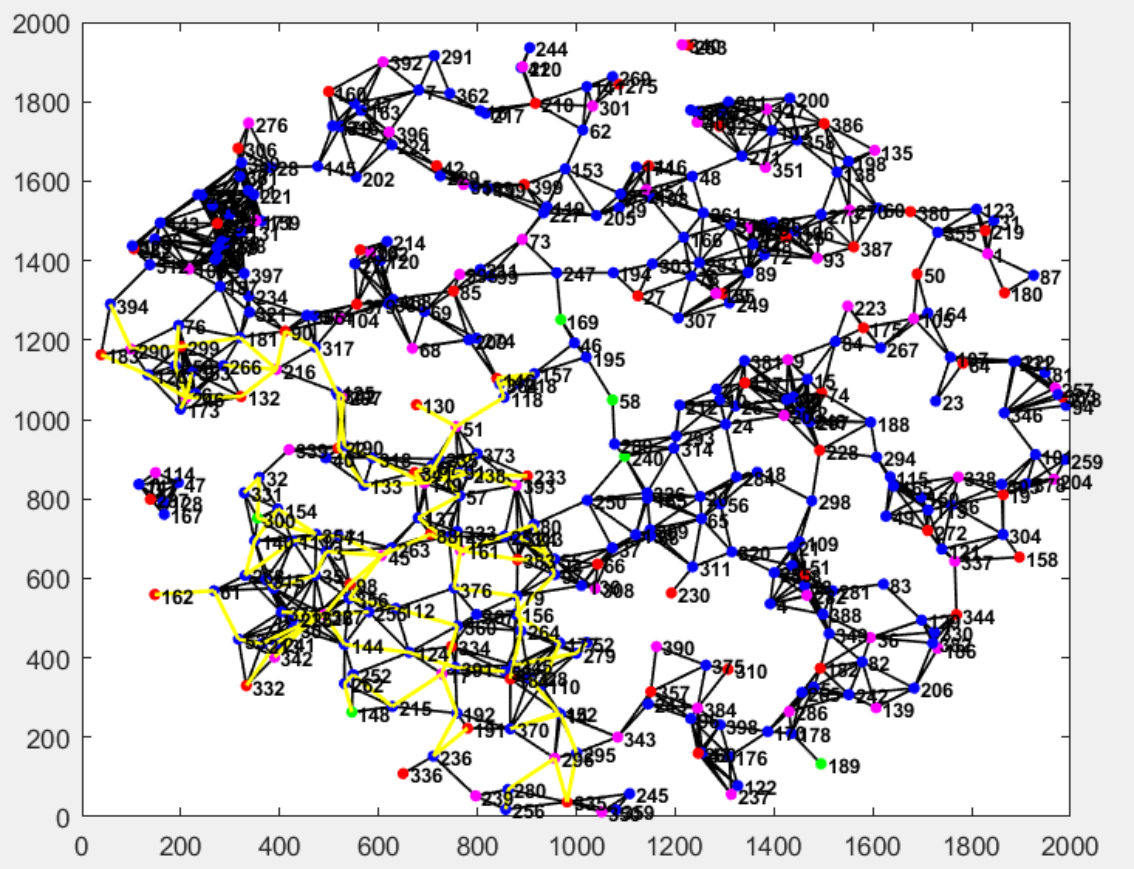

Image 4.17 Network evolution for one message – 20 minutes

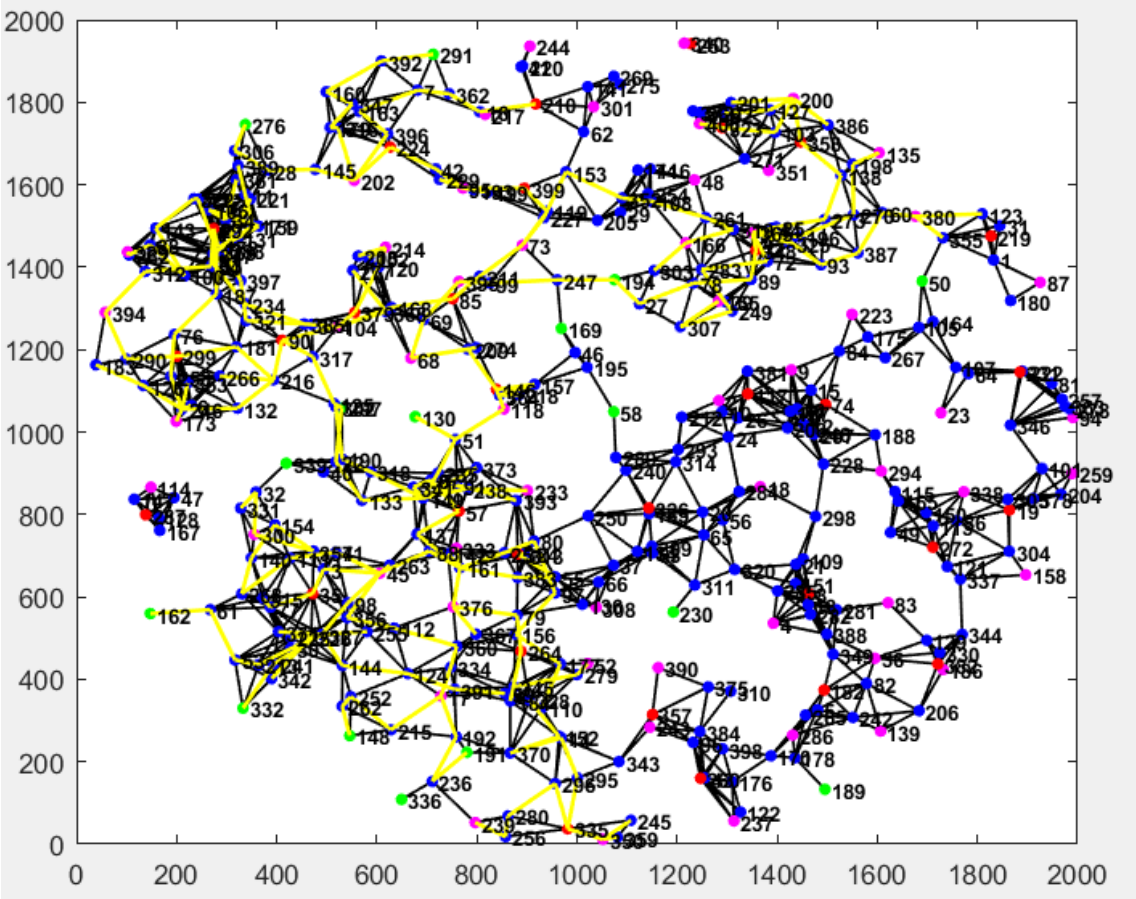

Image 4.18 Network evolution for one message – 40 minutes

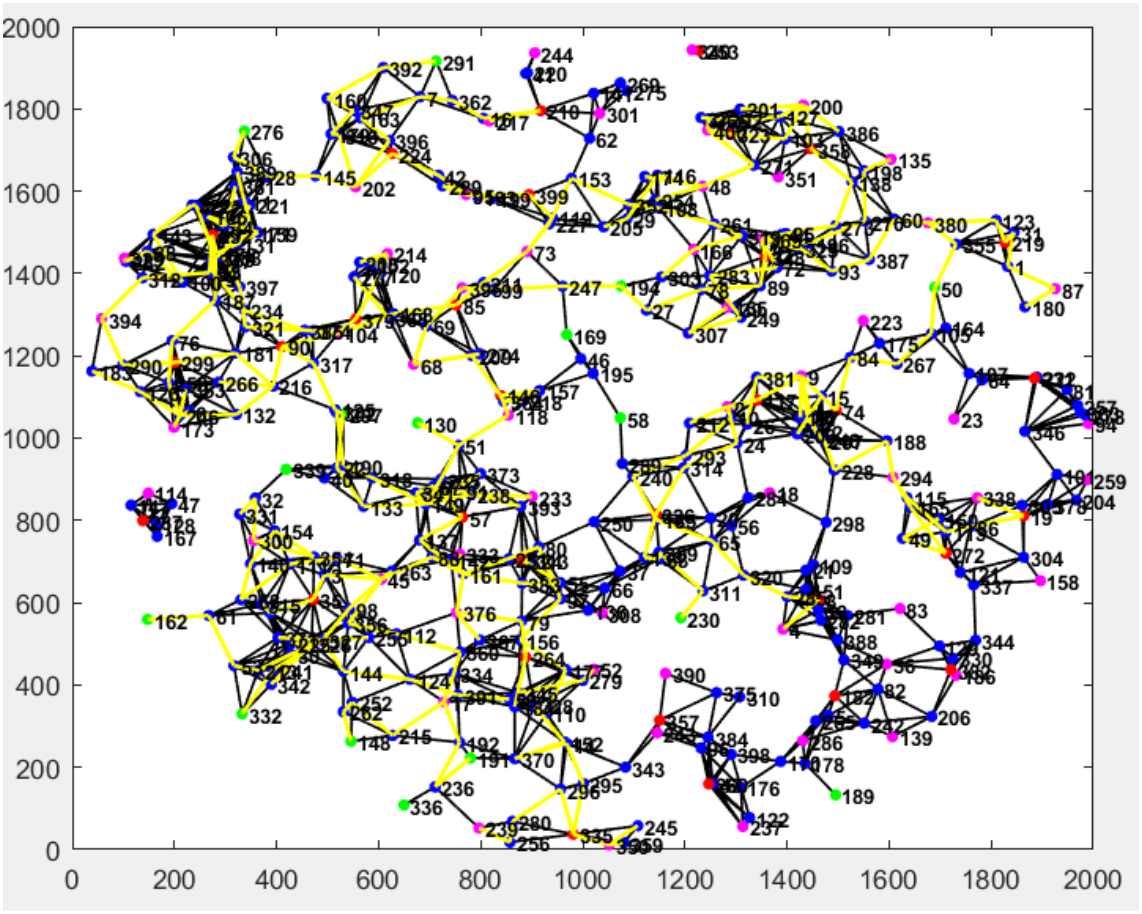

Image 4.19 Network evolution for one message – 60 minutes

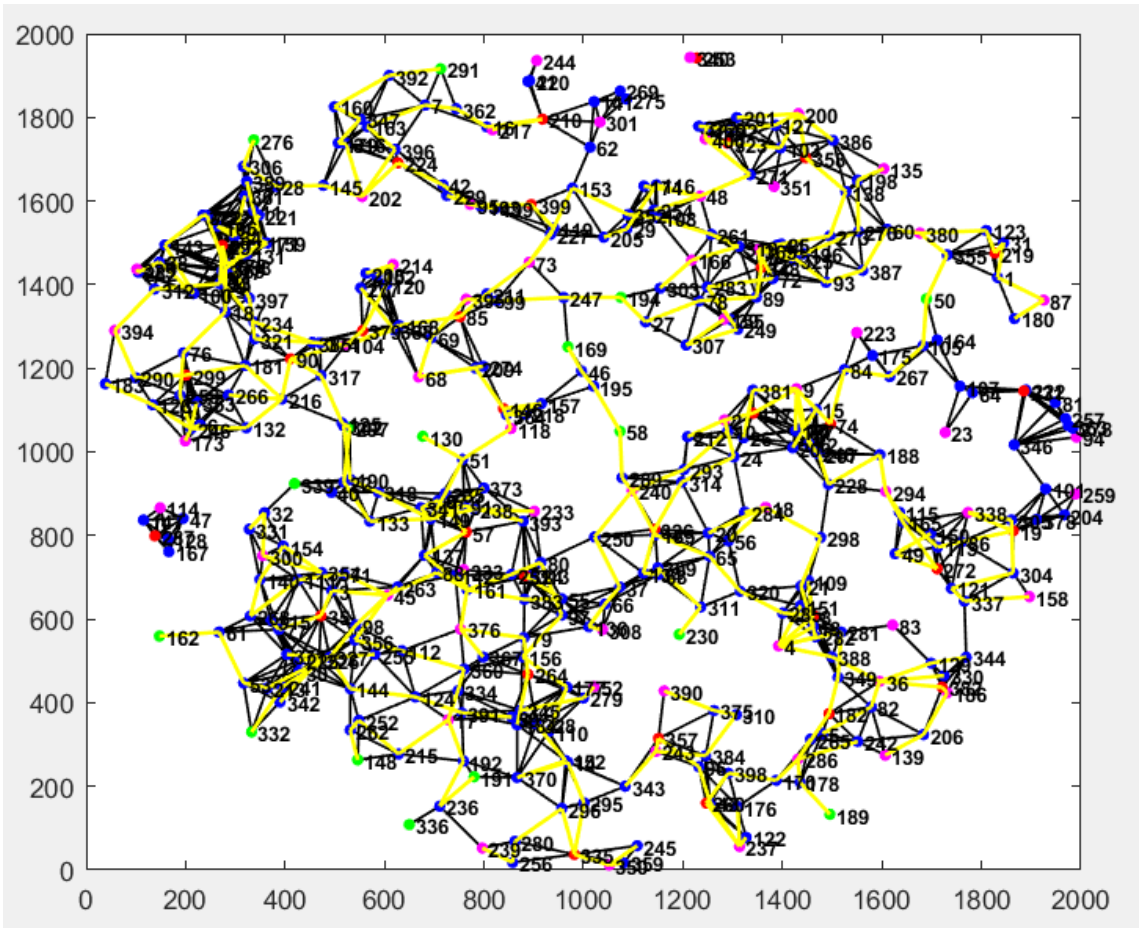

Image 4.20 Network evolution for one message – 80 minutes

4.6.2 Number of sent messages in relation with time

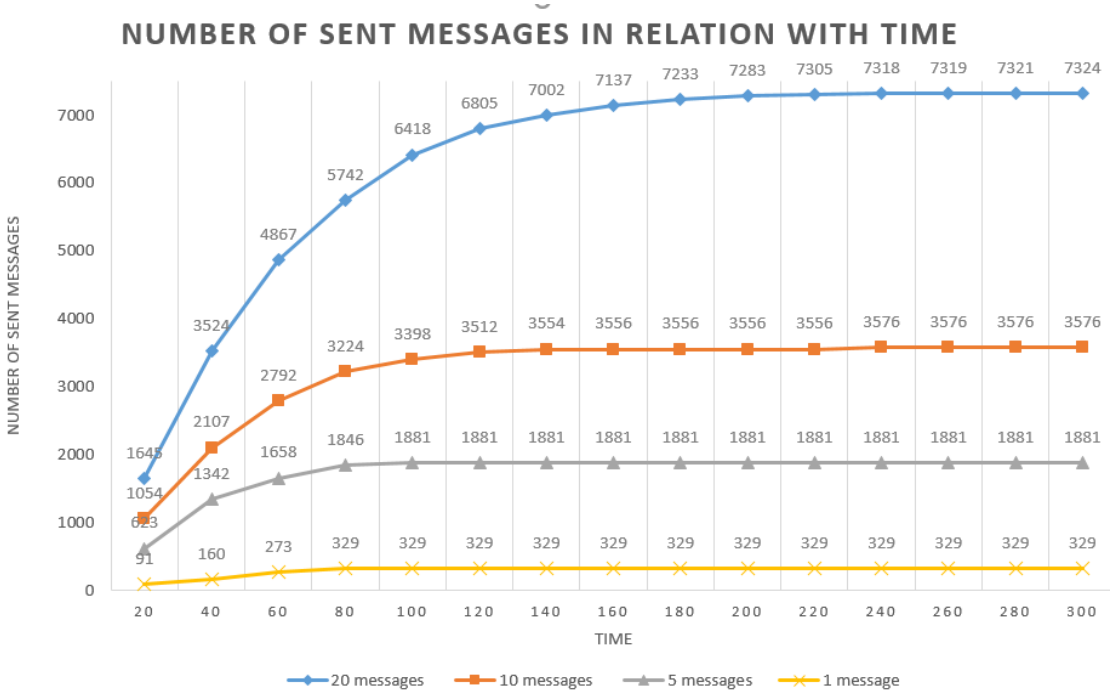

Image 4.21 number of sent messages in relation with time

At the graph is demonstrated the number of sent messages in relation with time in a network of 400 nodes. The four lines are representing the number of the nodes that send their message through the network. More specifically, the blue line is for 20 messages, the orange one for 10 messages, the gray line is for 5 messages and last the yellow line for 1 message.

The number of messages that are sent throughout the network is analogous to the nodes that have a message to send. In the network where are sent messages from 20 nodes the final number of sent messages is 7324. As the number of the nodes that have a message to send is decreasing the messages sent from all the nodes are decreasing too, since for the 10 messages the number of the messages sent is 3576. Also for the duration of the 300 minutes if the sent messages are 5, the number is getting lower at the 1881 messages and for the one message the number of sent messages is 329.

Additionally, is obvious that the nodes stop the dissemination of the messages earlier if the number of the messages that are travelling in the network is smaller. The yellow line is stabilizing at the 80 minutes, the gray line at the 100 minutes, the orange line at the 260 minutes and final the blue line becomes stable after the 300 minutes.

Number of sent messages	1 message	5 messages	10 messages	20 messages
100 minutes	329	1881	3398	6418
	329/1 = 329	1881/5 = 376.2	3398/10 = 339.8	6418/20 = 320.9
200 minutes	329	1881	3556	7137
	329/1 = 329	1881/5 = 376.2	3556/10 = 355.6	7137/20 = 356.85
300 minutes	329	1881	3576	7324
	329/1 = 329	1881/5 = 376.2	3576/10 = 357.6	7324/20 = 366.2

Table 4.30 Number of send messages in relation with rescue messages

4.6.3 Percentage of cluster heads that received the messages of all the nodes in relation with the time

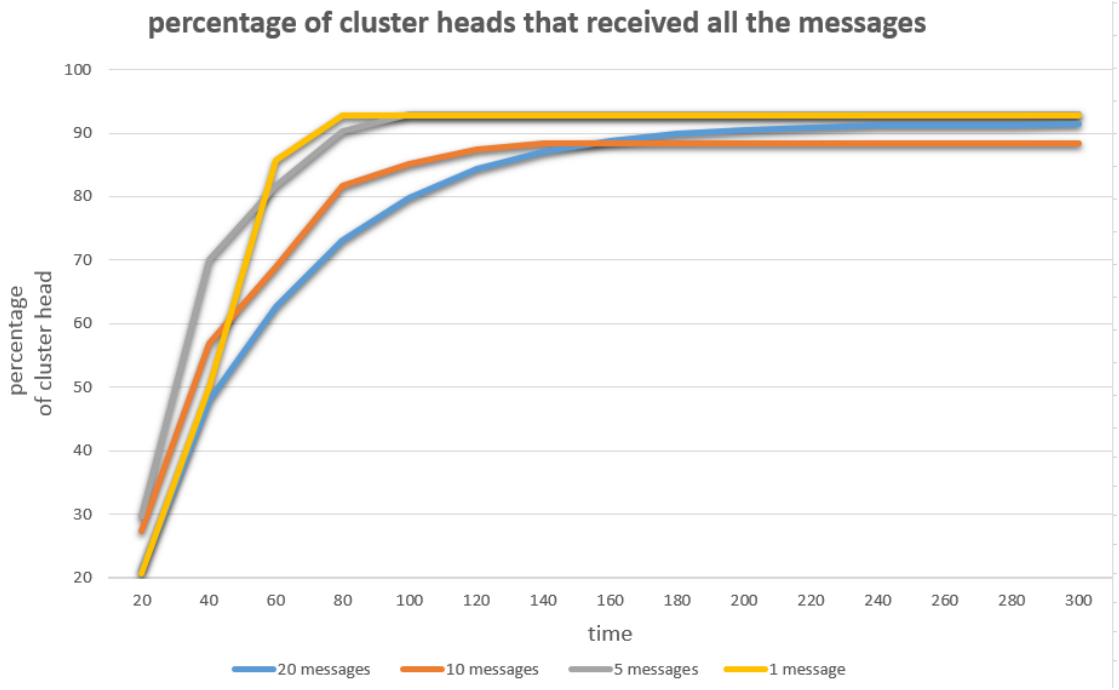

Image 4.22 percentage of cluster heads that received all the messages in relation with time

The graph illustrates the proportion of cluster heads that received the messages of all the nodes that are sent to the network versus the time. The graph is showing four different cases with the four lines representing different number of nodes that have a message to send through the network.

For all the cases the percentage of cluster heads that received the messages from all the nodes is approximately 90 per cent. The difference between them is the period of time that passed in order to all the messages end up at the 90 per cent of the networks cluster heads.

As the number of nodes that have a message to send to the network increases the time needed to reach the preferable percentage of cluster heads is more.

The yellow line, that represents the network where only the message of one node is sent, reaches the 92.85% first comparatively with the other cases. After the gray line reaches the percentage of 92.94 cluster heads at the 100 minutes. The other two lines are reaching their final proportion of cluster heads after some time passes, since the red line is stabilizing at 180 minutes and the blue line at 300 minutes.

4.6.4 Percentage of bridge nodes that received the messages of all the nodes in relation with the time

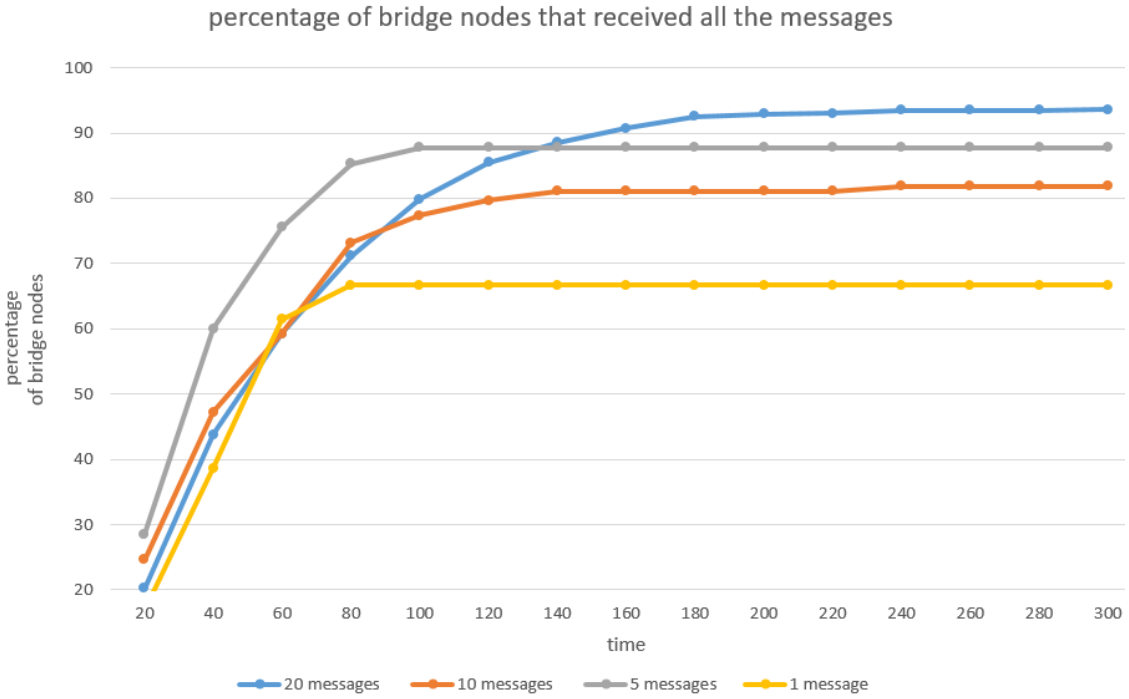

Image 4.23 percentage of bridge nodes that received all the messages in relation with time

The graph shows the percentage of bridge nodes that received the messages of all the nodes in relation with time.

The four lines are representing:

- A network with 20 nodes sending their message (blue)
- A network with 10 nodes sending their message (orange)
- A network with 5 nodes sending their message (gray)
- A network with 1 node sending it message (yellow)

The time that pass in order to reach the final proportion of bridge nodes is analogous to the number of nodes that are sending their message to the network. In the cases with the less messages, the time, that is needed from the final percentage of bridge nodes to receive the messages from all the nodes is less than the others.

4.7 Conclusion

In summary of all our scenarios we may conclude in some overalls inferences. Firstly, the results of the scenario C comparatively with scenario A are similar. This leads us to the conclusion that the change that has been made is negligible.
Secondly, the results of the last scenario for the battery decrement are lower from the other scenarios. Basically, the alternation of the scenario D it was intended to reserve the power of the device. The cost of this change, is obvious at the results of the other metrics. Such as the percentage of the cluster heads that received the messages of all the nodes, which is lower than the others scenarios
Finally, the change that is made in scenario B aim to decrease the overhead of the signals send in the signaling operation. The decreasing of the signals was accomplished but the messages send during the whole duration of the simulation between the nodes were more for the second scenario.

Chapter 5

Conclusion

We have managed to perform an evaluation of the explore and exploit algorithm in emergency response networks. The algorithm achieves the main goals we expect from it. The main goals of the algorithm are the longevity of the network and receive the message of each survivor a large amount of cluster heads.

In order to achieve the longevity of the network we used the cycles, for some duration the devices are in sleep mode and save power. Furthermore, the key for the second goal are the signaling technique to discover the bridge nodes and the routing flag which helps the messages to follow a path and not deviate from it.

From the evaluation we may conclude that our algorithm is consistent.

Bibliography

[1] Al-Sherbaz A., Dravid R., Svennevik E., Picton P. (2012). iSurvival: A Collaborative Mobile Network System for Disaster Management.

[2] Franck Legendre, Theus Hossmann, Felix Sutton, Bernhard Plattner. (2011). 30 Years of Wireless Ad Hoc Networking Research:What about Humanitarian and Disaster Relief Solutions?What are we still missing?

[3] G Jayakumar, G Gopinath. (2007). Ad hoc mobile wireless networks routing protocols–a review.

[4] G Zussman, A Segall . (2003). energy efficient routing in ad hoc disaster recovery networks.

[5] *Matlab Support*. (n.d.). Retrieved from https://www.mathworks.com/support/?s_tid=gn_supp

[6] Panayiotis Kolios, Andreas Pitsillides, Osnat Mokryn. (2013). Bilateral Routing in Emergency Response Networks.

[7] Panayiotis Kolios, Andreas Pitsillides, Osnat Mokryn, Katerina Papadaki. (2014). Explore and exploit in wireless ad hoc emergency response networks.

[8] Panayiotis Kolios, Andreas Pitsillides, Osnat Mokryn, Katerina Papadaki. (2014). Qualifying Explore and Exploit for Efficient Data Dissemination in Emergency Adhoc Network.

[9] Val Jones, Georgios Karagiannis, Sonia Heemstra de Groot . (2005). Ad hoc networking and ambient intelligence to support future disaster response.

[10] Varun G Menon, Joe Prathap P M. (2016). Analysing the Behaviour and Performance of Opportunistic Routing Protocols in Highly Mobile Wireless Ad Hoc Networks .

[11] Wenrui Zhao, MostafaH. Ammar . (2003). Message ferrying: proactive routing in highly-partitioned wireless ad hoc networks.