Macrothink Institute™

Network Protocols and Algorithms
ISSN 1943-3581
2018, Vol. 10, No. 3

Scatternet Formation Protocol for Environmental Monitoring in a Smart Garden

Lorena Parra[1], José Marín[2], Pedro V. Mauri[3], Jaime Lloret[4], Virginia Torices[5], Alberto Massager[6]

[1, 3] Instituto Madrileño de Investigación y Desarrollo Rural, Agrario y Alimentario, Finca El Encin, Autovía del Noreste A-2, Km. 38.200, 28805 Alcalá de Henares, Madrid

[1, 4] Instituto de Investigación para la Gestión Integrada de Zonas Costeras (IGIC), Universitat Politècnica de València (UPV). C/ Paranimf, 1, 46730 Grau de Gandia, Gandia

[5, 6,] Universidad Politécnica de Madrid. Escuela Técnica Superior de Ingeniería Agronómica, Alimentaria y de Biosistemas. Av. Puerta de Hierro, 2, 28040 Madrid

[2] Areaverde MG Projects SL. C/ Oña, 43 28933 Madrid

[1]loparbo@doctor.upv.es, [2]jmarin@areaverde.es, [3]pedro.mauri@madrid.org, [4]jlloret@dcom.upv.es, [5]virtorices@gmail.com, [6]alberto.masaguer@upm.es

Received: July 26, 2018 Accepted: October 29, 2018 Published: October 31, 2018

DOI: 10.5296/npa.v10i3.14122 URL: https://doi.org/10.5296/npa.v10i3.4122

Abstract

The monitoring of different parameters in the smart garden environment requires thousands of nodes and actuators. They form a multi-hop communication network. The scatternets formed with Bluetooth protocol is a communication solution. However, there is no current algorithm that considers the different capabilities of the devices (sensors or actuators) and assigns a role according to these capabilities. In this paper, we present a network topology formation algorithm for role assignment and connection establishment which considers the capabilities of the devices and use slave-slave Bridge to communicate the piconets. We design the algorithms needed for this protocol and test it. We have simulated the algorithms in order to evaluate the time needed for role assignment and to establish the first connections of the piconet. The results include different scenarios composed by one or two masters and one to seven slaves. In addition, we evaluate the established connections in piconets and bridges in a real case of the smart garden sensor network. Finally, we present the changes in the piconet connections after the deployment of two nodes in an existing network.

Keywords: Bluetooth, formation algorithm, WSN, smart garden, SS-bridge

1. Introduction

Smart city concept is becoming more and more popular. Many applications are being developed under this wide concept. The final objective of the smart cities besides with the Internet of Things (IoT) is to provide internet connection to the common objects. Their purpose is to improve the efficiency of the processes and enhance the sustainability of the city [1]. The majority of the applications for smart cities are composed by nodes embedded in different objects [2] with sensors and/or actuators and a system that triggers the actuators according to the data gathered by the nodes. Different options are available, since the most straightforward, acting when a threshold is overcome; to more complex ones, involving the combination of factors in Artificial Intelligence (AI) systems. The aforementioned concept can be applied for monitoring public areas, like gardens, where different supplies are consumed as water and electricity. Moreover, other resources are needed for maintaining the different infrastructures of the gardens such as fountains, grass, trees, artificial lakes and recreational areas among others. In addition, it is possible to monitor the environmental parameters to evaluate the welfare of the people in the parks and gardens. The wireless sensors networks (WSNs) can be used for monitoring all the parameters to provide data to our system. We can find examples for garden monitoring [3], the use of IoT for smart metering [4], or for human welfare monitoring [5], our objective is to join them.

The main problem of this wide WSNs in public gardens is the difficulties to communicate the nodes. On the one hand, they are far away from each other, separated by tens of meters. This, can difficulty to use WiFi to communicate them. WiFi is probably the most employed technology for WSN. Nonetheless, the available WiFi antennas for Arduino nodes have coverage of tens of meters [6]. This is insufficient to communicate the nodes in big gardens. Therefore, another technology should be selected as Bluetooth which have a range of up to 100 meters [7]. On the other hand, as the public gardens in big cities can have large dimensions, a multi-hop communication is needed. Thus, the energy consumption for communication issues become critical in some nodes, which should resend the data from many nodes [8]. A good solution is to use of a mobile node that will act as a sink to gather the data from all the deployed sensors in the garden [9]. However, this creates a secondary problem. If we use WiFi with the mobile node, we should use Bluetooth, and the capacity of the scatternets, to connect the rests of the sensors to allow them to send urgent messages.

The aim paper is to propose a new protocol for scatternet formation. Our algorithm needs to ensure that the sensor nodes (SN) set as masters and the actuator nodes (AN) set as slaves. In addition, the algorithm has to ensure that each actuator forms part of the piconet of the closest sensor. We will design the need algorithms for the formation of the scatternet. Moreover, we present the evaluation of this protocol in terms of time for the piconet establishment in different scenarios. Furthermore, the creation of piconets and scatternets in a portion of the smart garden is presented.

The rest of the paper is structured as follows. Section 2 shows the related work. The proposal is detailed in Section 3. Section 4 summarizes the performed simulation and show the results. The conclusions and future work are detailed in Section 5.

2. Related work

In this section, the related work is presented. We show different proposals of formation protocols for scatternets in Bluetooth networks and other group protocols of networks. Then, we explain why the current solutions are not useful for our proposal of WSN in the smart garden.

T. Hassan et al. [4] in 2008 presented a solution for the scatternet formation problem, the Ring of Masters (ROM). The authors show a new ring structure for Bluetooth scatternets with dynamic routing and adaptive scheduling schemes. The novelty of their proposal was the new decentralized ring structure, which is aimed to combat the bottlenecks. Their protocol allows flexibility of node selection and a good level of fault tolerance. In their protocol, they use slave-slave bridges to connect the piconets. Nonetheless, the ring topology is not appropriated for the expected topology in the urban gardens. Y. Li et al. [5] presented the Bluetooth Growing Network (BGN) algorithm for scatternet formation in 2011. Their formation algorithm was specific for WSN. Their algorithm was addressed to solve the problems of unreliability and inefficiency of the current algorithms for Bluetooth scatternet formation. Their results point that their algorithm was efficient attending to the number of created bridges, data transmission distance, etc. However, in their algorithm all the nodes are SNs, no ANs are included in the network. Therefore, each node has the same characteristics and can become master or slave. Even if their algorithm is efficient, it cannot be applied to solve our problem in the network of the smart garden.

In 2012 S. Sharafeddine et al. [6] presented a scatternet formation algorithm for Bluetooth networks with a non-uniform distribution of devices. The authors presented a new scatternet formation protocol called BlueHRT (Bluetooth Hybrid Ring Tree). Their protocol has different stages including discovery, role assignment, interconnection of piconets to create the scatternet, and routing protocol. The authors propose to use the via slave–slave and master-slave bridges to connect the piconets. This proposal is not useful in our case. The role assignment and the interconnection of piconets to create the scatternet do not accomplish the requirements established in the introduction. However, the rest of the protocol (discovery and routing) suits our requirements. Therefore, we pretend to modify the role assignment and piconet connection algorithms and use the BlueHRT modified protocol.

The idea of having a network with devices that have different characteristics shown in 2003 by D. Reading-Picopoulos and A. A. Abouzeid. In their paper [7], they define high power devices that can be slaves or masters and low power devices, which can be slaves or masters. Nonetheless, they do not use the different type of device to assign the role. Similar paper can be found in 2003, where C. Pamuk and E. Karaúan [8] considered the different Device Grade to assign the role. The algorithm assumes that the master of the node must have a higher Device Grade. The Device Grade is defined as a combination of the battery capacity, battery level, and traffic generation rate properties. This protocol uses master-slave bridges.

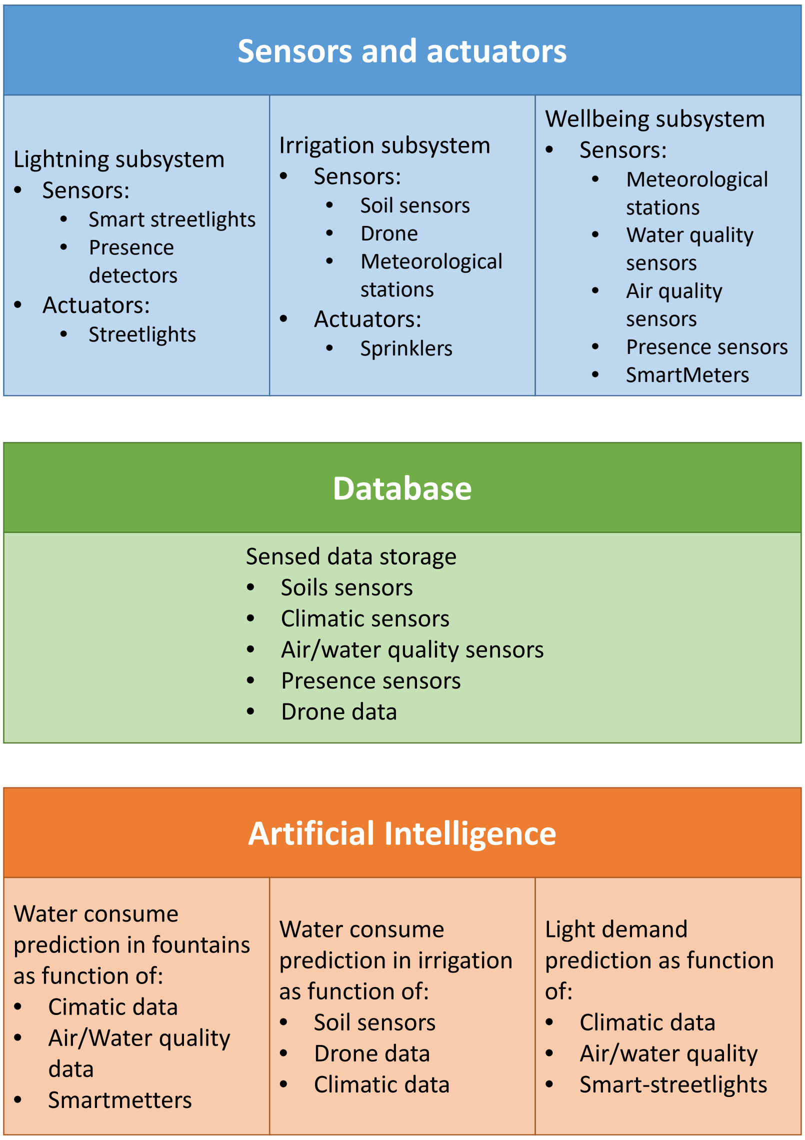
As far as we know there is no one protocol that considers the different characteristics of nodes to assign a role, which uses the slave-slave bridge to join piconets.

3. Proposed system

In this section, we present the proposed system including the architecture, the employed sensors, the topology and the proposed protocol for scatternet formation.

3.1 Architecture

The system for the smart garden is composed of three different layers. The first layer contains the nodes that can be sensors or actuators. This is the layer in charge of gathering data from the environment and performing actions. The second layer is the database where all the data is stored far away from the garden. This layer is composed of different databases to ensure that the information can be recovered even if there is a failure. The last layer is an artificial intelligence system, which operates with the data from the database. The artificial intelligence system is used to predict future behaviors based on analyzing the data from the previous events. The architecture can be seen in Fig. 1.



Sensors and actuators

Lightning subsystem
- Sensors:
  - Smart streetlights
  - Presence detectors
- Actuators:
  - Streetlights

Irrigation subsystem
- Sensors:
  - Soil sensors
  - Drone
  - Meteorological stations
- Actuators:
  - Sprinklers

Wellbeing subsystem
- Sensors:
  - Meteorological stations
  - Water quality sensors
  - Air quality sensors
  - Presence sensors
  - SmartMeters

Database

Sensed data storage
- Soils sensors
- Climatic sensors
- Air/water quality sensors
- Presence sensors
- Drone data

Artificial Intelligence

Water consume prediction in fountains as function of:
- Cimatic data
- Air/Water quality data
- Smartmetters

Water consume prediction in irrigation as function of:
- Soil sensors
- Drone data
- Climatic data

Light demand prediction as function of:
- Climatic data
- Air/water quality
- Smart-streetlights

Figure 1. Architecture of the smart garden.

The system is divided into three subsystems (SuS). The first SuS is in charge of the irrigation issues of the garden. Therefore, it has sprinklers as actuators, and as sensors, it has a soil moisture sensor and a soil temperature sensor. In addition, a drone is used to gather pictures of the grass to evaluate its wellness**.** The data from the soil (humidity and temperature) is stored in the node. The data is sent to the mobile sink twice a day. The drone takes pictures, which are combined with the data from the soil sensor. Both data are jointly used by the AI system to decide which sprinklers should be activated. The sprinklers can irrigate at sunrise and at sunset. The drone and the soil node can send a message to the sprinklers to modify the scheduled irrigation. The messages sent by the node to the sprinklers are the result of the AI system. On the other hand, the messages send by the soil nodes to the sprinklers indicates that the values of soil parameters had exceeded a threshold. Therefore, it is necessary to increase the amount of water in the next irrigation period.

The second SuS is the lightning SuS and it manages the streetlights of the public garden. It is composed of sensors and actuators as the previous one. As a sensor we have two types of smart streetlights. The first type of smart streetlights have a light sensor. These sensors are responsible of sending a message to the actuators to turn on the other lights, but not all of them. The second type of smart streetlights have a presence sensor. During the night, those sensors are responsible for activating the rest of the streetlights when it is needed. The streetlights that have no sensors act as actuators turning on and off the light according to the received messages from the smart streetlights. All the data gathered by the smart streetlights is stored and transmitted to the mobile sink when it is available. Nevertheless, if the gathered data exceeds the established thresholds in terms of lack or excess of light and detection of people, messages are sent from one node, smart streetlight, to the rest of the streetlights.

The last SuS is the welfare SuS. In public gardens, people spend their time: Nonetheless, they are exposed to inclement weather, air pollution and even water pollution among others. It is important to monitor the quality of life of people in the garden. In the park, there are a lot of kids playing and elderly people. Those people are two of the risk groups of people that should be more aware of environmental circumstances. For this reason, it is important to monitor the environmental parameters that can affect the quality of life in smart garden. The SuS of welfare is composed of sensors and actuators. The sensors that are part of this SuS are climatic sensors, air quality sensors, water quality sensors, and sound sensors. In addition, this system will use the data from the smart meters of the public fountains and the presence sensors to monitor the number of people that are in the garden. On the other hand, the actuators of this subsystem are visual indicators which show the quality of life of people in the park. These indicators can be used by people to decide to remain or not in the park. The data gathered by the different utilized sensors is stored in the nodes. They wait for the pass of the sink node to send all this information. Besides, if one of the parameters overcome the established maximum or minimum values a message is sent to actuators.

3.2 Topology

In this subsection, the topology of our network is presented. First, we explain the structure of our network in terms of utilized nodes and the roles of each node. Then, we explain the connections between nodes.

In the network for the smart garden, we have three big sorts of nodes according to their role. There are some nodes that have the capability to sense the environment. Those are the sensor nodes. They have to measure different parameters and are in charge of store data and send it if it is necessary. They provide data to the AI to take the appropriate actions. There are different types of sensor nodes according to the subsystem and to the parameter that the sensor measures. In the light subsystem, there are two types of sensor nodes, the ones that measure the light intensity and the ones that measure the presence of people. In the irrigation, the subsystem has just one type of sensor nodes. Those sensors, measure the soil parameters including the soil temperature and the soil humidity. Finally, for the subsystem of the quality of life, the system has six different types of sensors. There are four types of sensors for monitoring the environment such as climatic sensors, air quality sensors, water quality sensors and sound sensors; and two types of sensors that monitories the human activity in the garden such as smart meters and presence sensors. All the sensor nodes have two antennas, one of them to work as Ad-Hoc mode and the other to work in Infrastructure mode.

The second sort of node are the actuators. Those nodes do not measure any parameter. Their role is related to the reception of data from the sensor nodes and triggers the actuator. There are less actuator node types than sensor node types. There is only one type of actuator in each SuS, the actuators are the streetlights, the sprinklers, and the visual indicators. The actuator nodes have only one antenna which operates in Ad-Hoc mode.

Finally, there is one node that will be mobile, which have the role of downloading the data stored in the sensor nodes, it is also called gateway. Moreover, the gateway node should send messages to the actuator nodes through the sensor nodes. The gateway node has only one antenna. This antenna operates under Infrastructure mode.

Now, the topology of the network is described. It can be seen in Fig. 2 and Fig. 3. In Fig. 2 we showed the possible location of different types of sensor nodes in one portion of the urban garden. The different colors and shapes represent different SNs types. There are a total of 8 types of SNs and 3 types of ANs. Following, Table 1 indicates the different types of SNs and ANs. In Fig 2 we can see a portion of the urban garden that contains all the types of SNs. The ANs are not represented because there are plenty of them including the streetlights, with a density in the paths of 4 streetlights each 100m^2. In Fig 3 we can see the deployment of some ANs and SNs in the garden and their aspect.

3.3 Communication technology

In order to communicate our nodes, we are going to use two different technologies for two purposes. In our network, there are two situations when the nodes will need to communicate each others. The first one is when an SN needs to send alarm messages to an AN. This communication must be fast. For example, when the SN that measures the light intensity detects that is getting dark. It must send a message to some of the AN indicating the need of turning on the lights. Even if we can have a delay from a couple of minutes, we need to communicate the nodes at this moment. We will use the Bluetooth technology to send this sort of data.

On the other hand, twice a day it is necessary to transmit all the data to the database. In this case, we are going to transmit a large quantity of data in a short time. The drone will be in charge of collect all the data from each SN. The communication proposed for this purpose is the IEEE 802.11g.

3.3 Scatternet formation

In this subsection, we detail the issues related to the scatternet formation and the Bluetooth issues, which should be considered. First of all, we detail some ideas that must be considered in our formation protocol. Then, we detail the algorithms that form part of the protocol.

Table 1. Description of the SNs and ANs of the smart garden.

SN		AN	
Type	**Description**	**Type**	**Description**
1	Soil sensor	1	Sprinkler
2	Presence sensor	2	Streetlight
3	Water quality sensor	3	Welfare indicator
4	Air quality sensor		
5	Climatic sensor		
6	Noise sensor		
7	Light sensor		
8	Smart meters		

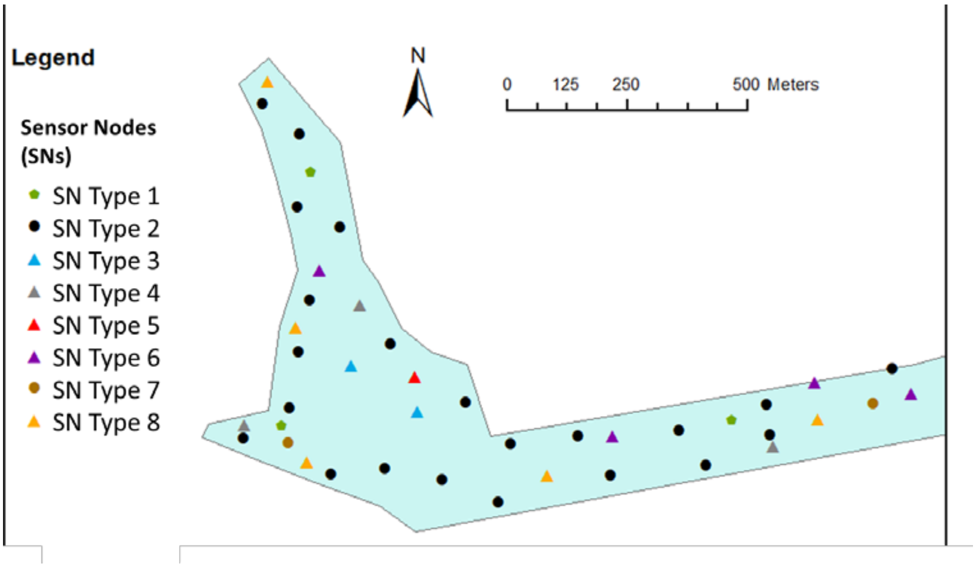


Figure 2. Location of SNs in a portion of smart garden.

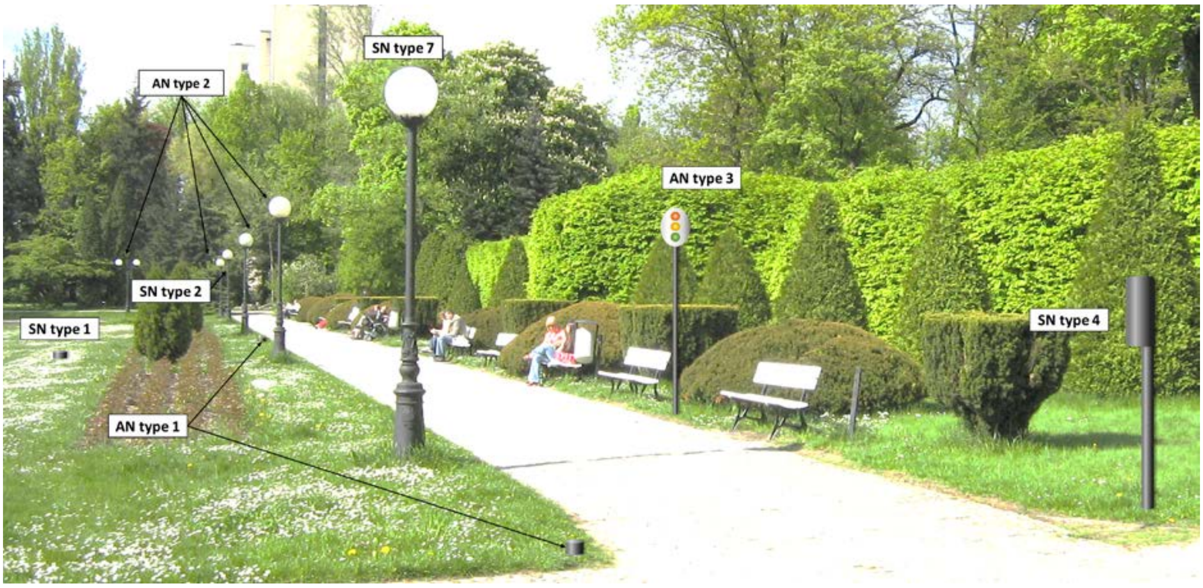


Figure 3. Deployment and identification of ANs and SNs in the smart garden.

According to the operation of Bluetooth communication, we can have nodes with the role of master and nodes with the role of slave. The communication is always started by the master and two slaves cannot communicate directly. A single master can communicate with 7 slave nodes creating a piconet. One node can belong to more than one piconet with or without the same role. The nodes that belong to two different piconets are responsible for connecting both piconets. This type of connections is known as scatternet. In our network, we will have some nodes, the SN, that sends messages (alarms) to the other nodes, the AN, which are only receiving data and answering with the ACKs. These different behaviors can be perfectly implemented in the piconets where our SNs become the masters and the ANs are the slaves. Moreover, we have more AN than SN, what will allow connecting multiples ANs to a single master.

The first step to operate in a Bluetooth network with multi-hop topology is the formation of the scatternet. There are several algorithms that allow the formation of the Ad-Hoc scatternet. The basis of most of them is that each node starts to alternate between two states known as inquiry and inquiry scan. In the first state the node sends inquiry requests and in the second state, the node is listening for inquiry requests. When a node is listening and receives an inquiry request replies sending an inquiry reply and a piconet is formed. The node that sent the request takes the role of master, and the one that sends the reply takes the role of slave. Initially, a node has the same possibilities to end as a master or a slave in the piconet.

In order to create the scatternet, it is necessary to connect piconets. There are three options to connect piconets. The different options are represented in Fig. 4. In the first option, master of the piconet A becomes the slave of master of piconet B, see Fig 4 a). Then one node has two roles in two different piconets . It is known as SM-bridge. This is the option with less efficiency. An intermediate option is the creation of a third piconet connecting two slaves from piconet A and piconet B, see Fig 4 b). The best option in terms of efficiency is to share a common slave. Therefore, a slave from one piconet, for example, piconet A is connected to the master of piconet B, see Fig. 4 c). This is the best option and the shared slave is known as SS-bridge. Different terms can be considered to select the node that will act as SS-bridge as the first responding node, remaining energy or distance to masters.

For our system, the current algorithms are not useful, because it has some special requirements. Firstly, we need to establish that the SNs act as Masters always. The ANs act as a slave is possible and only will take the role of master if there is no piconet available for the AN. To solve this, we can assume that initially when the nodes are deployed we can indicate to each node if it is a SN or an AN, and include in the inquiry packet the type of node. However, it supposes that during the deployment we must indicate the type of node. Our nodes only know in which SuS are working (lightning, irrigation or wellbeing). Initially they do not know their role in the net. Nevertheless, there is a difference between the SN and the AN, the level of initial energy. As the SNs need to measure environmental parameters, send alarms via the Bluetooth network and the data via the WiFi network, they need to have higher energy level. All the nodes will have an energy harvesting system that ensures a long lifetime. Moreover, in normal conditions, even if a SN consumes more energy it is expected that always will have more remaining energy than the AN.

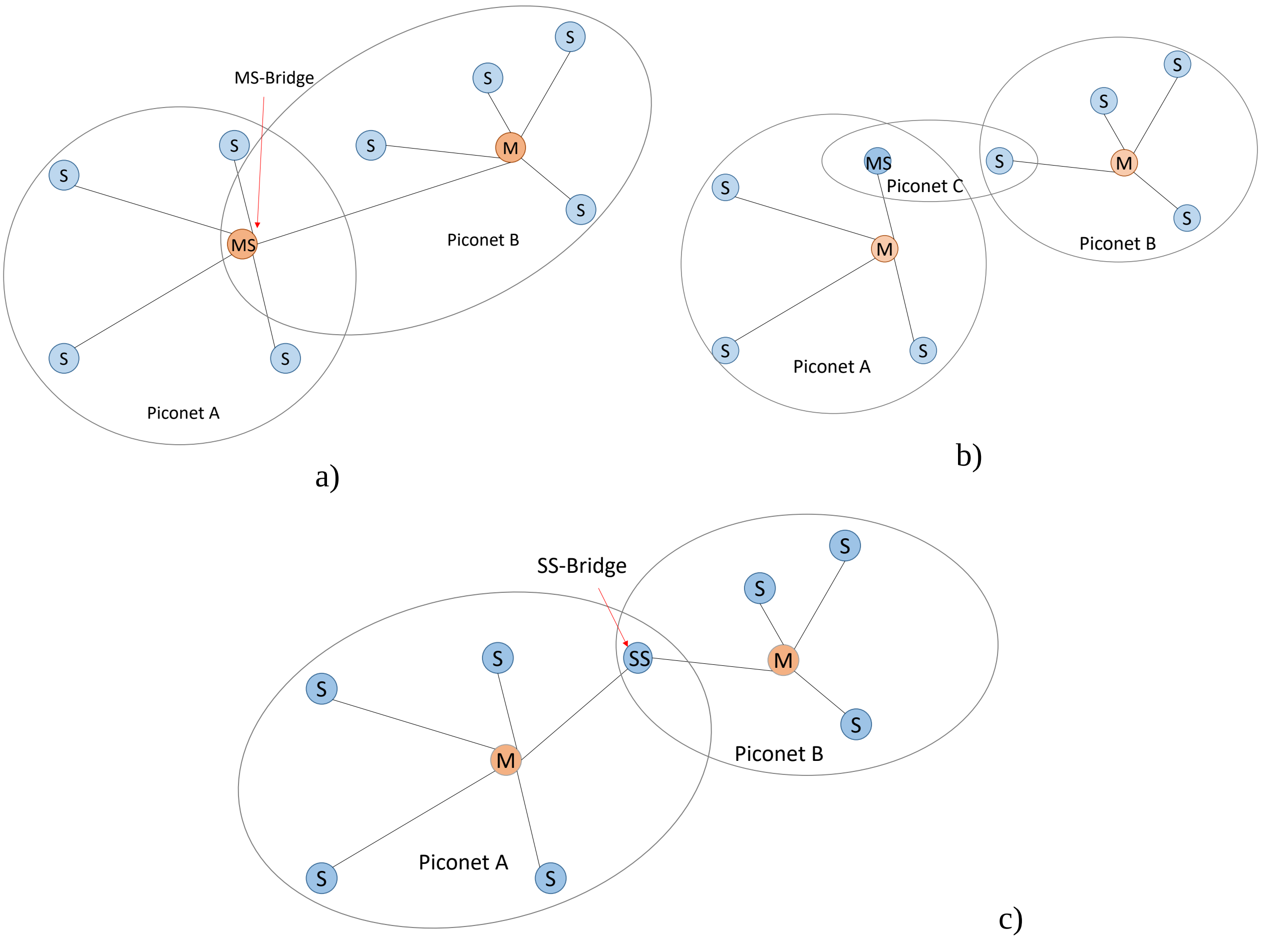


Figure 4 Different options for scatternet formation

Therefore, the energy level can be included in the inquiry message and used to determine if the message comes from a different type of node (SN or AN) and known the type of node that receives the message. Thus, a connection can be created and the master and slave are defined. Besides, if the energy level received in the inquiry is similar to the energy level of the node that receives the inquiry it is not possible to know the type of nodes. They only know that both of them are the same type of node and they cannot establish a connection.

Another requirement is that the slave must be connected to the closest master. Therefore, during the formation phase, there will be some freedom to change from one master to the other in order to ensure that each slave is connected to the closest master. If one slave is equally distant from two masters, the slave will connect to the master which is part of the same SuS (lightning, irrigation or wellbeing). During this first stage, a slave can only connect to one master.

The algorithms used during this stage 1 are the following ones. Fig. 5 shows the first stage of phase 1, in this stage, each node will discover if it has the role of master or slave. Each node selects a random number between 0 and 10, this is the time that the node will remain in inquiry or in scan inquiry mode. The minimum time is 625uS, which is the time that needs one node to send the inquiry message in all the frequencies. According to the random number, the node will start emitting or scanning.

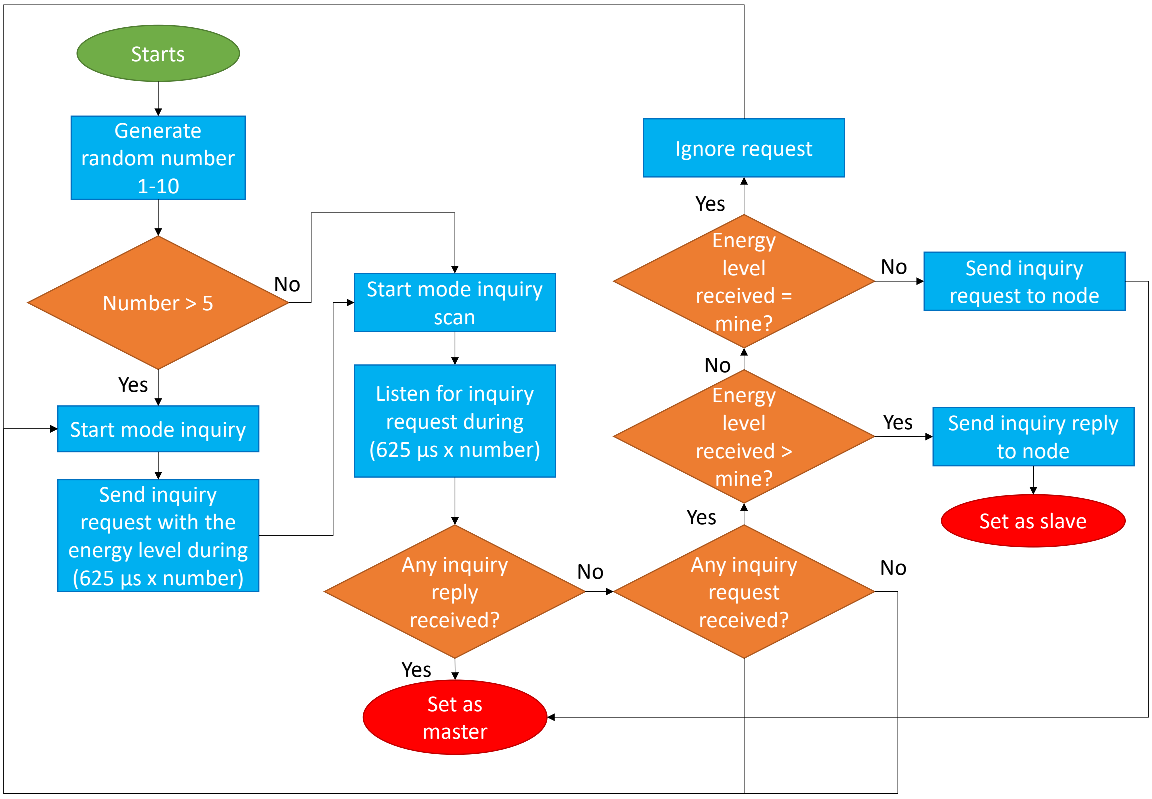

Starts
Generate random number 1-10
Number > 5
No
Yes
Start mode inquiry
Send inquiry request with the energy level during (625 µs x number)
Start mode inquiry scan
Listen for inquiry request during (625 µs x number)
Any inquiry reply received?
Yes
No
Set as master
Ignore request
Yes
Energy level received = mine?
No
No
Energy level received > mine?
Yes
Yes
Any inquiry request received?
No
Send inquiry request to node
Send inquiry reply to node
Set as slave

Figure 5. Algorithm 1 Phase 1, the role assignment.

When a node is scanning mode it scans only one frequency each time period. Once they discover their role in the network the node applies the algorithm of Fig. 6 if the node is a master or the algorithm of Fig. 7 if the node is a slave. The master uses 11 slots of time. Thus, the master ensures that all the slaves listen to its message when it is emitting and listens to all the messages from the slaves when the node is scanning. The master will continue sending inquiry request and scanning for inquiry replies in order to find closest slaves that belong to the same SuS.

In the case of slaves, they generate a new random number and keep scanning in order to find other possible masters that are closer to the node and belong to the same SuS. The objective of this phase 1 is to ensure that each master is connected to its closest slaves. This phase ends when no more changes are detected.

The phase 2 stats once all the slaves are part of the piconet of the closest master. Then, it is the moment to from the scatternet, and it is necessary to create links between piconets. During this phase, the slave can create a new link with a second master as it is explained above. The objective of the algorithms in Fig. 8 and Fig. 9 is to find the closest slave from the neighboring piconets. A master which has already 7 slaves finishes the phase two. This phase finish when no more new links are done.

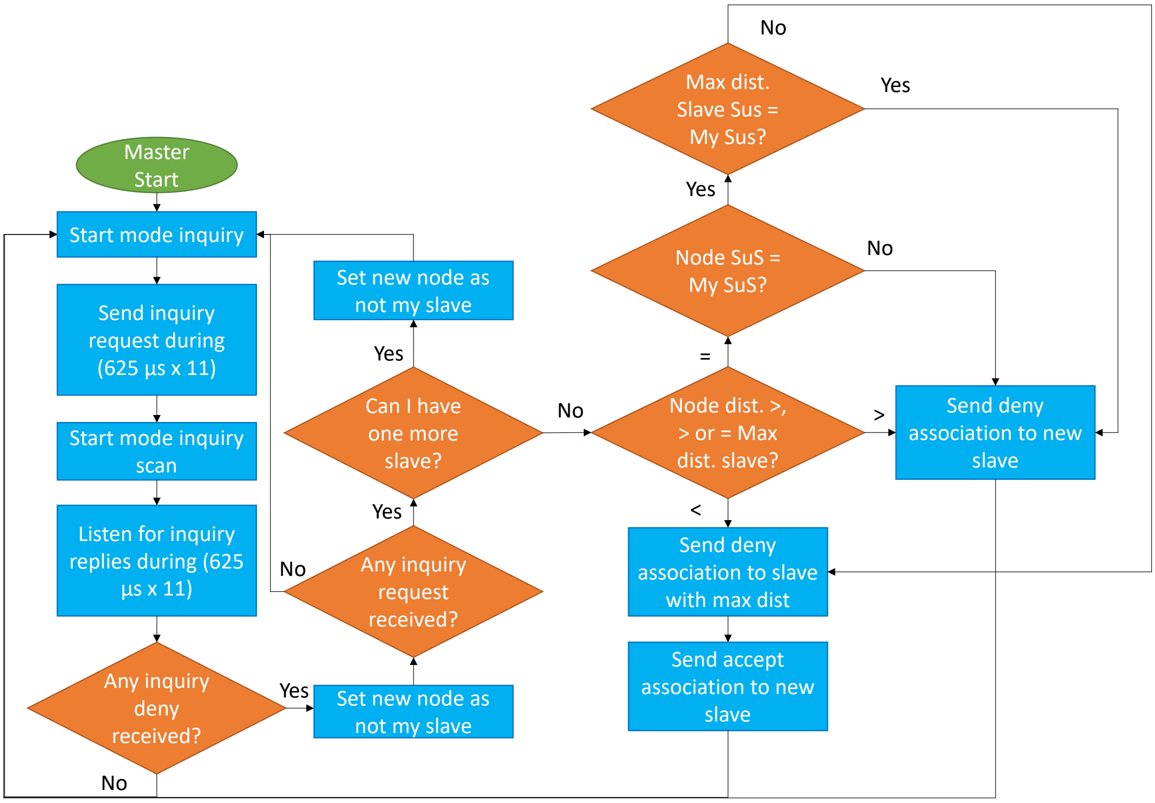

No
Max dist. Slave Sus = My Sus?
Yes
Master Start
Yes
Start mode inquiry
Set new node as not my slave
Node SuS = My SuS?
No
Send inquiry request during (625 µs x 11)
Yes
=
Can I have one more slave?
No
Node dist. >, > or = Max dist. slave?
>
Send deny association to new slave
Start mode inquiry scan
Yes
<
Listen for inquiry replies during (625 µs x 11)
No
Any inquiry request received?
Send deny association to slave with max dist
Any inquiry deny received?
Yes
Set new node as not my slave
Send accept association to new slave
No

Figure 6. Algorithm 2 Phase 1, the master operation pattern.

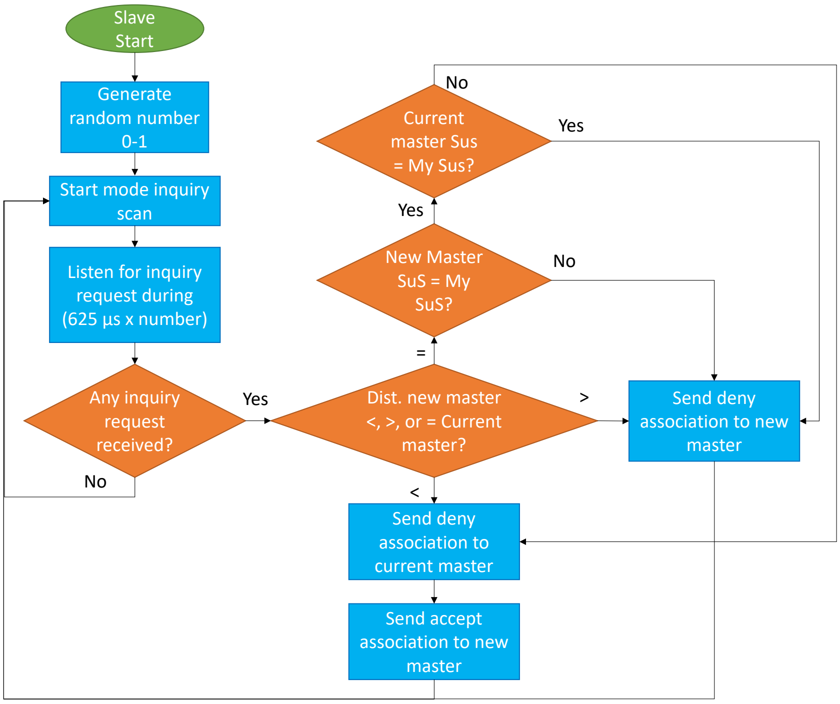

Slave Start
Generate random number 0-1
Start mode inquiry scan
Listen for inquiry request during (625 µs x number)
Any inquiry request received?
Yes
No
Dist. new master <, >, or = Current master?
>
=
<
New Master SuS = My SuS?
Yes
No
Current master Sus = My Sus?
No
Yes
Send deny association to new master
Send deny association to current master
Send accept association to new master

Figure 7. Algorithm 3 Phase 1, the slave operation pattern.

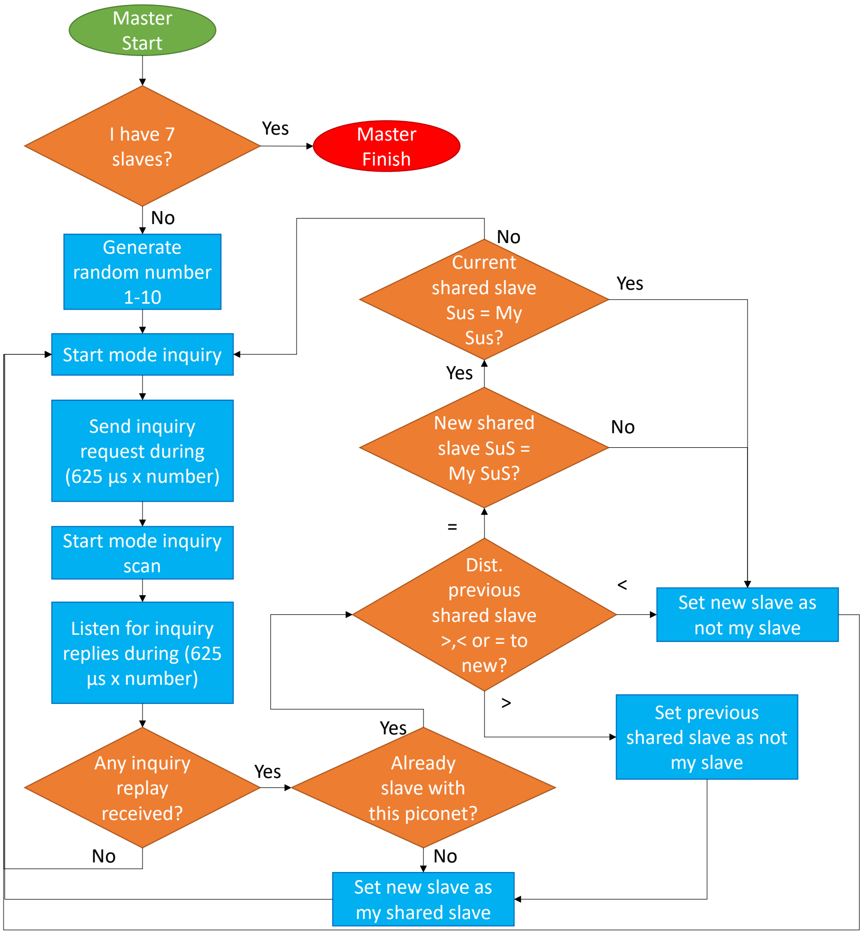

Master Start
I have 7 slaves?
Yes
Master Finish
No
Generate random number 1-10
Start mode inquiry
Send inquiry request during (625 µs x number)
Start mode inquiry scan
Listen for inquiry replies during (625 µs x number)
Any inquiry replay received?
Yes
No
Already slave with this piconet?
Yes
No
Set new slave as my shared slave
Dist. previous shared slave >,< or = to new?
<
>
=
New shared slave SuS = My SuS?
Yes
No
Current shared slave Sus = My Sus?
No
Yes
Set new slave as not my slave
Set previous shared slave as not my slave

Figure 8. Algorithm 1 Phase 2, the master operation pattern.

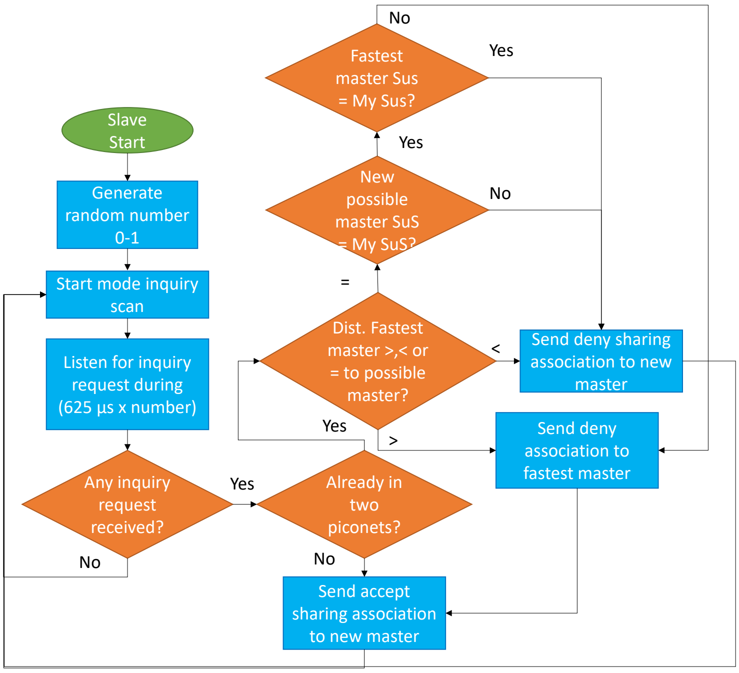


Figure 9. Algorithm 2 Phase 2, the slave operation pattern.

Finally, we have to consider the formation that will happen when a new node arrives into an established network. This is known as phase 3. When a new node arrives, it has to scan searching existing masters to join to previously created piconets. If no master is available because all of them have 7 slaves, then the new node will become a master. Otherwise, it will become a slave of the closest master with space in the piconet for one more slave, see Fig. 10.

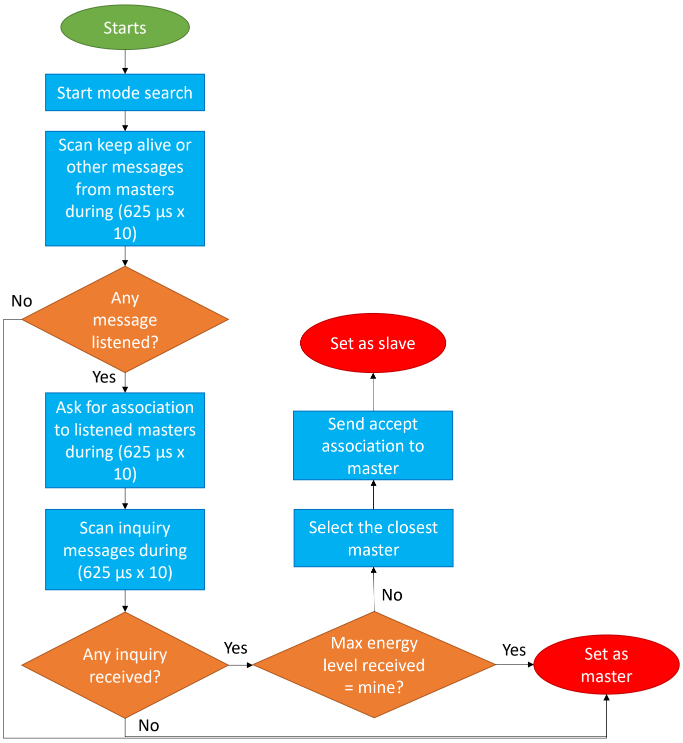


Figure 10. Algorithm 1 Phase 3, the operation pattern of a new node.

4. Results

In this section, we present the obtained results after simulating the proposed algorithms for role discovery and connections establishement. Firstly we show the time needed to discover all the roles in the network and the establishment of first connections. Then, we present for our smart garden, the piconet and scatternet formation and the changes after the introduction of two ned nodes a SN and a AN.

Now, we detail the results of the simulation to obtain the time needed for the formation of the piconet. Different simple scenarios are simulated. We have two cases, a case with only one SN and a case with two SN. For each case, we consider the effect of increasing the number of AN, from 1 AN to 7 ANs. We select 7 as maximum ANs, because in the case with one SN no more than 7 slaves can be connected. In addition, we differentiate if the SN starts in emitting mode (SE) of scanning mode (SS) in the case of one SN. In the case of two SN, there is an extra possibility when each one starts in a different mode (SDM).

We begin with the data of the case with just one SN. We perform 20 simulations of each scenario. Fig. 11 presents the time that the SN needs to determine that it is the master of the piconet. The time is lower when the node starts in SS. In this situation, the first inquiry received indicates that the SN is the master when it compares the energy level of the received inquiry and its energy level. The greater the number of AN the lower the time. Fig. 12 shows the time needed by the first AN to set as slave. This time is higher when the SN starts in SS. Again, the greater the number of AN the lower the time. Finally, Fig. 13 shows the time needed by the last AN to set as slave. This parameter is affected only by the mode that takes the SN. When the SN starts in SS the time is higher than when the SN starts in SS. Apparentrly, there are no changes in this time when we increase the number of ANs.

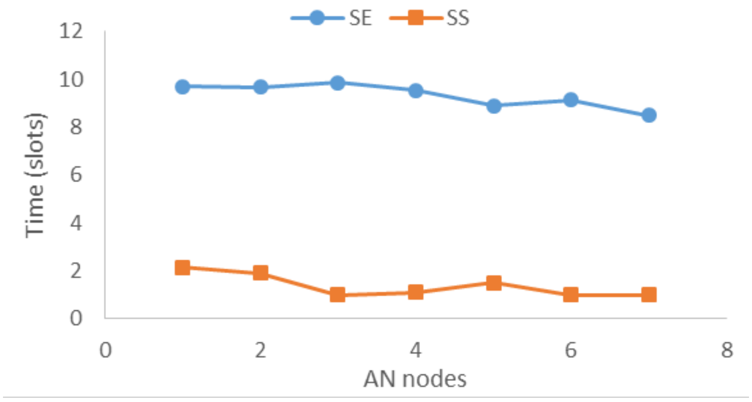


Figure 11. Time needed by the SN to set as master with one SN

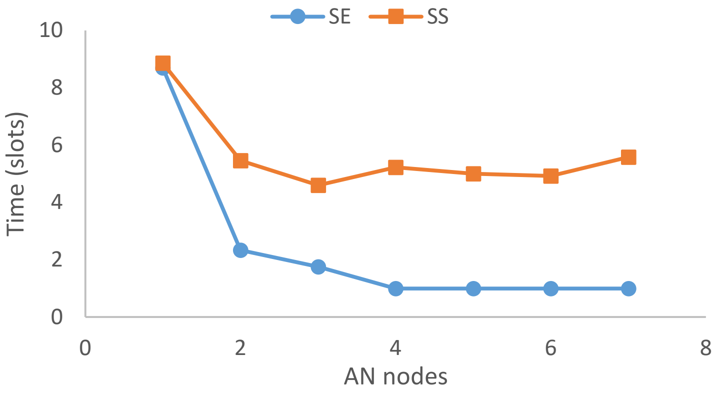


Figure 12. Time needed by the first AN to set as slave

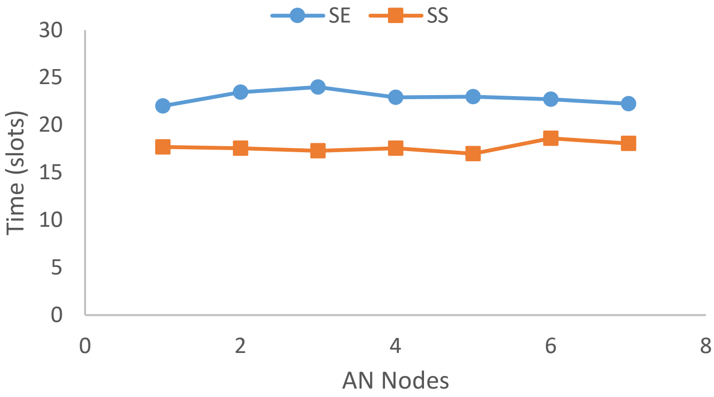


Figure 13. Time needed by the last AN to set as slave

In this paragraph, the results of the second case, having two SN and different number of ANs; are presented. As now there are three possibilities, SE, SS, and SDM; 20 simulations of each possibility have been done. Fig. 14 presents the time needed by the first SN to set as master. The results are similar to the ones shown in Fig 11. The time is greater when the SNs starts in SE. In addition, the time is reduced when the number of ANs is increased. The results are similar for SS and SDM when there are up to three ANs. Fig. 15 shows when the second SN set as master. In this case, the results for the SDM are a bit lower than the results for SE. The lowest times are related to the scenario when both SN starts in SS. Fig 16 presents the iteration when the first AN set as slave. As in Fig. 12, the starting mode of the SN is an important factor. The time is minimum when both SN starts in SE and maximum when both SN starts in SS. As in Fig. 12 the greater the ANs the lowest the time. The last evaluated parameter is the time needed by the last AN to set al slave, see Fig. 17. The results are similar to the results of Fig. 13. The time is a little higher when both SN starts in SE. In this scenario the time increases with the number of ANs.

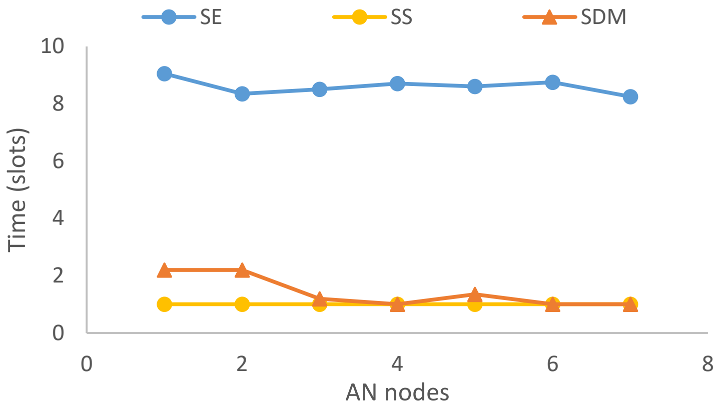


Figure 14. Time needed by the first SN to set as master with two SN

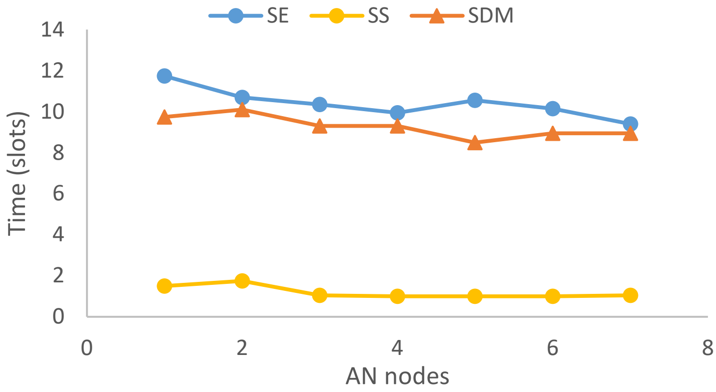


Figure 15. Time needed by the second SN to set as master with two SN

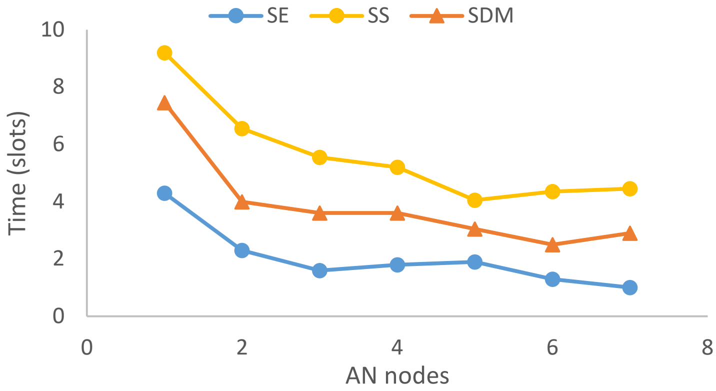


Figure 16. Time needed by the first AN to set as slave

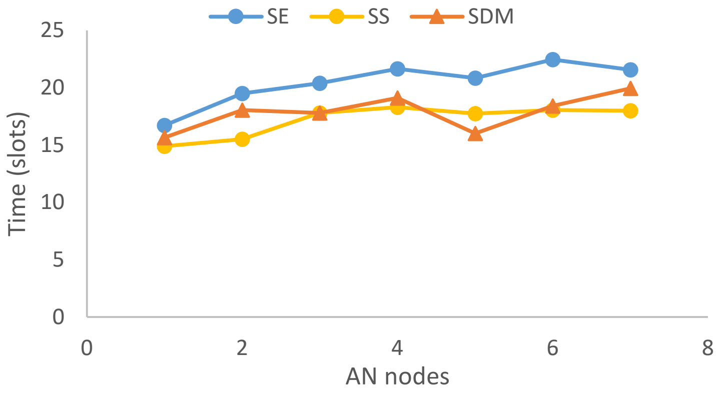


Figure 17. Time needed by the last AN to set as slave

Next, we present the formation results of the proposed algorithms in the case of the smart garden presented in Section 3. We consider a small section of the garden. This section includes 15 SNs and 60ANs. In this section, all the SuS and all the types of SNs and ANs are presented. It can represent a small scale smart garden. In blue color, we represent the illumination SuS, in green the irrigation SuS, and in other colors the welfare SuS. The squares represent the SN and the circles the ANs. In this portion of the garden there are different paths, gardened areas with grass coverage at both sides of the path, and a small water mass, an artificial lake, see Fig. 18.

The formation of the piconets after the application of algorithms presented in Fig 8 to 10 can be seen in Fig. 19. We can see that there are 15 piconets, because each SN is master of one piconet. The piconets of the center have 7 slaves and the piconets of the boundaries have fewer slaves (1 to 5 slaves). We can expect that if we increase the number of ANs all the SN will have 7 slaves. Thus, it exists the possibility that some central ANs should set as master.

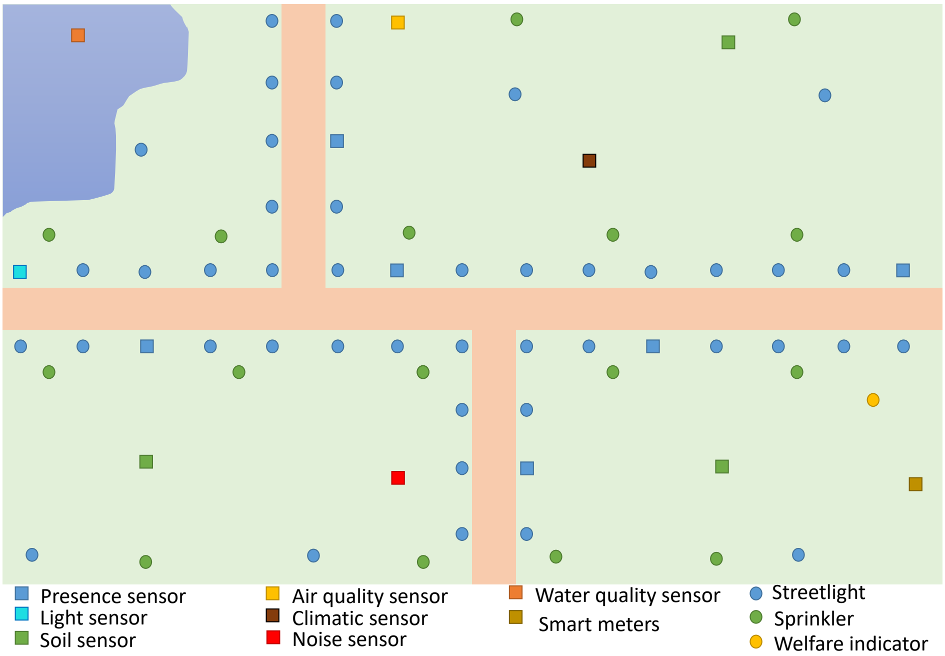


Figure 18. The portion of the smart garden utilized for simulation

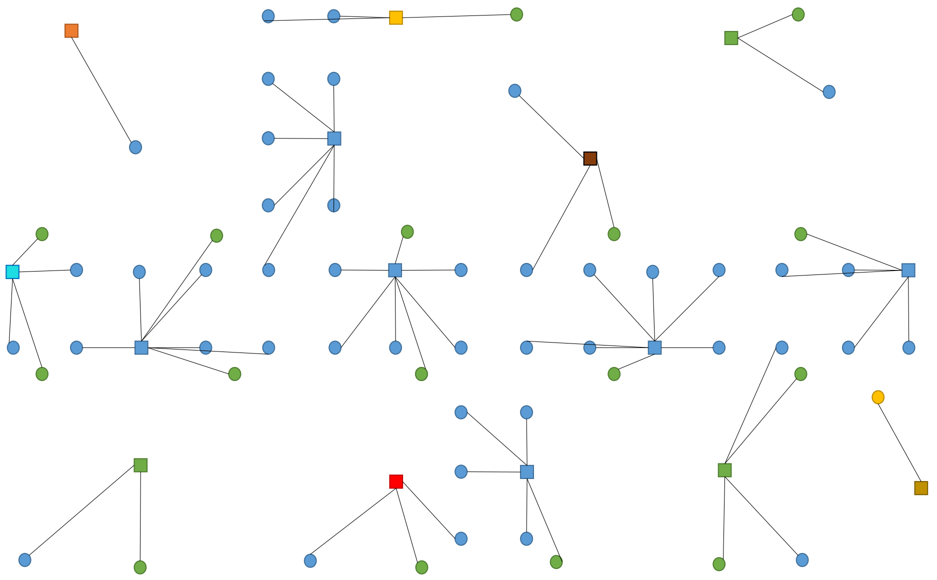


Figure 19. Piconets created after phase 1 in the portion of the smart garden

The results of phase 2, the creation of links between piconets, are presented in Fig. 20. A total of 23 new links have been created, which connects different piconets. No one piconet was isolated from the scatternet. The piconets of the boundaries are the ones that create more new links. One of them created four new links, connecting this piconet with four piconets. After the piconets connection, the available links are the ones presented in Fig. 21.

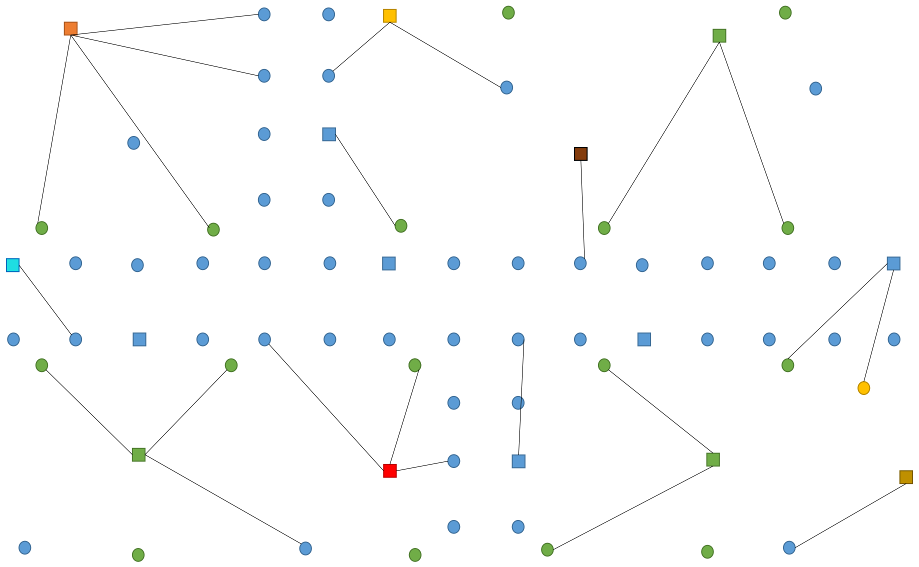


Figure 20. SS-bridge links created in phase 2

Finally, we present the application of phase 3, when two new nodes are deployed. We assume that a new AN and a new SN are deployed in the garden after the formation of the scatternet. The results can be seen in Fig. 22. The AN is connected to an existing piconet and the SN is set as master. In phase 2 the AN is connected to a secondary master offering a shortest SS-bridge between two piconets. This SS-bridge is shown in red and the deleted link to the shared slave appears in a dotted red line. In phase 2, the SN create four new links with shared slaves, creating four new SS-bridges.

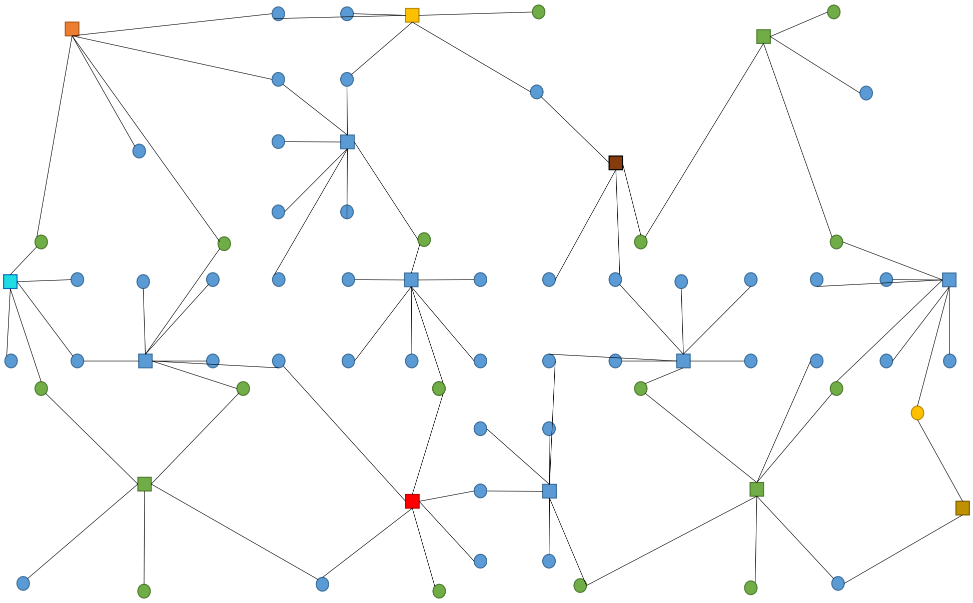


Figure 21. Scatternet formed after phase 2 in the portion of the smart garden

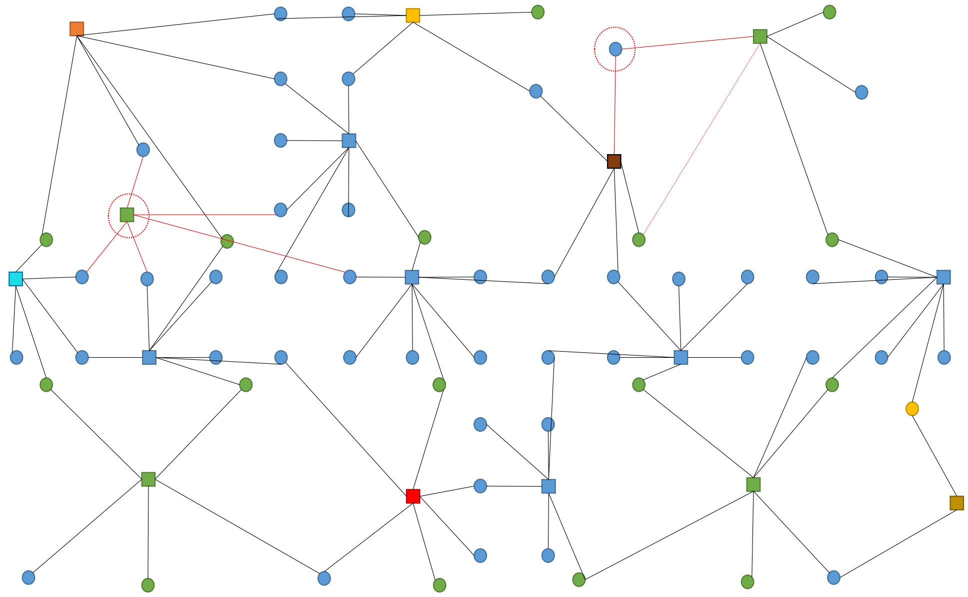


Figure 22. Scatternet formed after deploy two new nodes in the smart garden

4. Conclusion

In this paper, we analyze the needs of a network for urban garden monitoring. In these types of networks, we have a lot of ANs and SNs on land and drones flying to evaluate the state of the grass. This node can collect the daily data via WiFi. However, to send the alarms from the SN to the ANs a secondary connection is necessary. The best option to be combined with Wi-Fi is the Bluetooth, which can be used to create scatternets. Nonetheless, the existing protocols for scatternet formation consider that all the nodes are in equal conditions to become master or slave. For our proposal, we need that the SN set as master.

Taking everything into account, we present an algorithm for Bluetooth scatternet formation that considers the remaining energy as an indicator of the type of role. Furthermore, our algorithm can move the slaves from one piconet to another in the first phase to ensure that each slave is connected to the closest master. It also considers the different SuS, to connect a slave to the master of its SuS in case of the different master with the same distance are available. We show the results of simulating the proposed algorithm in terms of formation time considering different scenarios with 1 to 7 ANs and 1 to 2 SN. Moreover, we show the created links in a portion to the smart garden after phase 1, phase 2 and phase 3 after the deployment of two new nodes

As future works, we expect to evaluate the energy consumption when the alarms are sent from different SN as it was done in [15]. In addition, we pretend to include a phase 4 aimed to modify the SS-bridges in case that one of them reaches low energy levels.

Acknowledgement

This work is partially found by the European Union with the “Fondo Europeo Agrícola de Desarrollo Rural (FEADER) – Europa invierte en zonas rurales”, the MAPAMA, and Comunidad de Madrid with the IMIDRA, under the mark of the PDR-CM 2014-2020” project number PDR18-XEROCESPED. This work has been partially supported by the "Ministerio de Economía y Competitividad" in the "Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento" within the project under Grant TIN2017-84802-C2-1-P. This work has also been partially supported by European Union through the ERANETMED (Euromediterranean Cooperation through ERANET joint activities and beyond) project ERANETMED3-227 SMARTWATIR.

References

[1] J. Jin, J. Gubbi, S. Marusic, M. Palaniswami. An information framework for creating a smart city through internet of things. IEEE Internet of Things journal, 2014, vol. 1, no 2, p. 112-121. https://doi.org/10.1109/JIOT.2013.2296516

[2] A. Gharaibeh, M. A. Salahuddin, S. J. Hussini, A. Khreishah, I. Khalil, M. Guizani, A. Al-Fuqaha. Smart cities: A survey on data management, security, and enabling technologies. IEEE Communications Surveys & Tutorials, 2017, vol. 19, no 4, p. 2456-2501. https://doi.org/10.1109/COMST.2017.2736886

[3] J. Marín, L. Parra, J. Rocher, S. Sendra, J. Lloret, P.V. Mauri, A. Masaguer. Urban Lawn

Monitoring in Smart City Environments. Journal of Sensors, 2018, vol. 2018. https://doi.org/10.1155/2018/8743179

[4] J. Lloret, J. Tomas, A. Canovas, L. Parra. An integrated IoT architecture for smart metering. IEEE Communications Magazine, 2016, vol. 54, no 12, p. 50-57. https://doi.org/10.1109/MCOM.2016.1600647CM

[5] L. García, L. Parra, O. Romero, J. Lloret. System for monitoring the wellness state of people in domestic environments employing emoticon-based HCI. The Journal of Supercomputing, 2017, p. 1-25. https://doi.org/10.1007/s11227-017-2214-4

[6] J. Mesquita, D. Guimarães, C. Pereira, F. Santos, L. Almeida. Assessing the ESP8266 WiFi module for the Internet of Things. En 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA) 2018. p. 784-791. https://doi.org/10.1109/ETFA.2018.8502562

[7] HC Serial Bluetooth Products User Instructional Manual. Available in http://www.fecegypt.com/uploads/dataSheet/1480849570_hc06.pdf. Last access in 24/08/2018

[8] E. Fadel, V. C. Gungor, L. Nassef, N. Akkari, M. A. Malik, S. Almasri, I. F. Akyildiz. A survey on wireless sensor networks for smart grid. Computer Communications, 2015, vol. 71, p. 22-33. https://doi.org/10.1016/j.comcom.2015.09.006

[9] C. Tunca, S. Isik, M. Y. Donmez, C. Ersoy. Distributed mobile sink routing for wireless sensor networks: A survey. IEEE communications surveys & tutorials, 2014, vol. 16, no 2, p. 877-897. https://doi.org/10.1109/SURV.2013.100113.00293

[10] T. Hassan, A. Kayssi, A. Chehab. Ring of Masters (ROM): A new ring structure for Bluetooth scatternets with dynamic routing and adaptive scheduling schemes. Pervasive and Mobile Computing, 2008, vol. 4, no 4, p. 546-561. https://doi.org/10.1016/j.pmcj.2008.02.003

[11] S. Sharafeddine, I. Al-Kassem, Z. Dawy. A scatternet formation algorithm for Bluetooth networks with a non-uniform distribution of devices. Journal of Network and Computer Applications, 2012, vol. 35, no 2, p. 644-656. https://doi.org/10.1016/j.jnca.2011.10.004

[12] Y. Li, H. Chen, R. Xie, J. Z. Wang. BGN: A novel scatternet formation algorithm for bluetooth-based sensor networks. Mobile Information Systems, 2011, vol. 7, no 2, p. 93-106. http:// doi.org/10.3233/MIS-2011-0112

[13] D. Reading-Picopoulos, A. A. Abouzeid. A bluetooth scatternet formation algorithm for networks with heterogeneous device capabilities. In Proccedigns of the International Conference on Information Networking. Springer, Berlin, Heidelberg, 2003. p. 295-305. https://doi.org/10.1007/978-3-540-45235-5_29

[14] C. Pamuk, E. Karasan. SF-DeviL: distributed Bluetooth scatternet formation algorithm based on device and link characteristics. En Computers and Communication, 2003. (ISCC 2003). Proceedings. Eighth IEEE International Symposium on. IEEE, 2003. p. 646-651. https://doi.org/10.1109/ISCC.2003.1214191

[15] L. Parra, J. Rocher, L. García, J Lloret, J Tomás, O Romero, M. Rodilla, S. Falco, M. T. Sebastiá, J. Mengual, J. A. González, B. Roig. Design of a WSN for smart irrigation in citrus plots with fault-tolerance and energy-saving algorithms. Network Protocols and Algorithms 10 (2), 95-115. https://doi.org/10.5296/npa.v10i2.13205

Copyright Disclaimer