Contents management algorithm for Ad Hoc networks bio-inspired in the quorum sensing utilized by gram negative bacteria

Jorge Ernesto Parra Amaris

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial
Bogotá, Colombia
2018

Contents management algorithm for Ad Hoc networks bio-inspired in the quorum sensing utilized by gram negative bacteria

Jorge Ernesto Parra Amaris

In fulfillment of the requirements for the degree of:
Master in Engineering - Telecommunications

Advisor:
Ph.D Jorge Eduardo Ortiz Triviño
Co-Advisor:
Ph.D(C) Henry Zarate Ceballos

Research line:Ad-Hoc networks
Research Group: TLöN

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial
Bogotá, Colombia
2018

It's the computer age. Nerds are in.

Willow Rosenberg - *Buffy the Vampire Slayer*

Aknowledgements

First of all I would like to thank God, everything is because of Him. After god I would like to thank to the next people for all the help and support they gave me during the development of my dissertation.

- My parents: Jorge Ernesto Parra Urrea and Yamile Amaris Molina, my brothers: Fredy Alexander Parra Amaris and Camilo Andres Parra Amaris.
- My research group, specially I would like to thank Henry Zarate Ceballos Ph.D(C) for all his effort, advices and constant support during the development of this study, all of my classmates, colleagues and my thesis advisor.

Abstract

In this dissertation an applied observational study is developed to validate through simulation a theoretical model based on the quorum sensing for multi-agent communication to manage files within an Ad-Hoc network. The simulation of the model was carried out using NS-3 simulator under two different scenarios and proper statistical techniques were used to analyze the response. Additionally in order to gain more knowledge about the model sensibility analysis were performed.

Keywords: Ad Hoc Networks, multi-agent system, contents management, quorum sensing, simulation, ns-3.

Resumen

En esta investigación se desarrollo un estudio observacional para validar por medio de simulación un modelo teórico inspirado en el quórum sensing para comunicación entre agentes con el fin de administrar archivos dentro de una red Ad-Hoc. La simulación del modelo fue llevada a cabo usando el simulador NS-3 bajo dos escenarios de prueba diferentes además se utilizaron técnicas apropiadas para analizar la respuesta del modelo. Adicionalmente para obtener mayor información sobre el modelo se realizaron análisis de sensibilidad.
Palabras Claves: Redes Ad-Hoc, Sistemas Multi-Agentes, administración de contenidos, disponibilidad de la información, quorum sensing, simulación, ns-3.)

Contents

List of Abbreviations

Abbreviations

Abbreviation	Description
$MANET$	Mobile Ad-Hoc Network
WMN	Wireless Mesh Network
WSN	Wireless Sensor Network
MAS	Multi-Agent System
QS	Quorum Sensing
AI	Auto Inducers
$B - agent$	Bacterial Agent
MRS	Marginal Rate of Substitution
IID	Independent and Identically Distributed
CRN	Common Random Numbers
$NS - 3$	Network Simulator 3

1. Preface

Unlike traditional networks, Ad-Hoc are able to run services in absence of any infrastructure or centralized control, each host in the network is autonomous, together with the other hosts, they are able to set the parameters to configure the network. This decentralized behavior expands the vision of these kinds of networks, because it is necessary for each host to cooperate with one another, to make up for the absence of any kind of external aid and keep the network services running; this allows to see Ad-Hoc networks as communities and the hosts that conform them as the members who cooperate with each other.

Due to the fact that the hosts that conform the network are mobile and have hardware limitations, data availability is lower in contrast to traditional networks. To solve this problem, replication techniques have been used to ensure data availability inside the network. Even though replication is a good technique to improve data availability. Besides, replication is a process that can consume network resources and must be done without affecting the performance of the network. Storage capabilities between the network devices may differ and some won't be able to store replicas of large or full sizes; but if contents instead are segmented, devices can store pieces of contents and assemble them when is needed.

To expand the vision of the Ad-Hoc networks, improve the availability of contents and make the network even more autonomous, a Multi Agent System along with bio-inspired behaviors can be employed. Since multi agent interaction can optimize tasks and solve problems and bio-inspiration borrows behaviors from nature that are adaptive and highly scalable. One behavior that is of interest for this research is the Quorum Sensing, employed by Gram-negative bacteria, since it allows them to synchronize their group behavior and act collectively. Provided these three systems (Ad-Hoc networks, Multi Agent Systems and bacteria) can share many similarities and act together in order to accomplish tasks, the next research question is generated: How to improve contents management inside Ad-Hoc networks using a bio-inspired Multi Agent System?.

To solve this question, it is proposed a bio-inspired methodology, in which agents communicate in a fashion similar to bacteria, to manage contents and ensure their availably. Also agents are endowed with mechanisms to make decisions. Finally a methodology to segment files is proposed as well.

This research is an applied observational study to understand how a novel communication strategy for a multi-agent system interacting over an Ad-Hoc network will perform under different conditions. The validation was carried out through simulations making performance tests to determine if the system behavior obeys to expectations, applying appropriate statistical analysis to the simulation output and using sensibility analysis to gain understanding about the scope, variability and limitations of the model regarding changes in its parameters.

This document is organized as follows: first, in chapter two there is an introduction about Ad-Hoc networks, their main features, challenges, content management and some replication techniques are reviewed. On chapter three, multi agents systems and bacteria communication are reviewed additionally A bio-inspired algorithm for a Multi Agent system is developed; on chapter four a decision-making mechanism based on microeconomics is proposed. On chapter five, file segmentation is revised and a stochastic methodology to divide a file into pieces is suggested. Chapter six is about the simulation, verification and validation of the model, through statistics tests and sensibility analysis. Finally on chapter seven, there can be found the conclusions and recommendations for future works.

1.1. Objectives of the research

1.1.1. General objective

Propose an algorithm to manage segmented contents inside an Ad Hoc network inspired in the Quorum Sensing communication mechanism employed by Gram-negative bacteria.

Specific objectives

1. Characterize the Quorum Sensing through an algorithm.
2. Implement the proposed algorithm over a MAS, in which the communication between the agents is based on the principles of the QS.
3. Propose a methodology to partition contents.
4. Employ an optimization algorithm to endow the agents with abilities to make decisions.
5. Simulate the proposed method under two test environments
6. Validate and verify the methodology using sensibility analysis and statistics processes.

1.2. Knowledge contribution

1. The first contribution is a communication algorithm for multi-agent communication.

2. A decision-making mechanism based on microeconomics concepts, considering the consumers theory.

3. A file partition technique founded on the truncated-geometric distribution.

4. Even though that is not a contribution as such, this research was developed under the correct methodology to validate observational studies using simulation.

1.3. Academic production

- 1 conference paper
- The beginning of the writing process of a book about Ad-Hoc network simulation in ns-3

2. Ad Hoc networks, basic concepts

Nowadays, the utilization of wireless mobile devices such as laptops, cellphones and tablets has been increasing; these types of devices have become more affordable and are equipped with medium to high hardware specifications. In addition, the number of Internet connections has also risen not only through wired mediums but through wireless and cellular networks as well. Actual users demand access to a variety of contents anytime, anywhere, for instance anyone can check the latest discounts and sales while shopping at the supermarket on their cellphones. Today, there are lots of multimedia and content sharing applications available for different platforms, mobile computing and application development have become emergent markets, which has raised the interest of multiple companies.

A type of network that can take advantage of the alternatives brought by the growth in wireless communication technology and its applications, are the Ad Hoc networks [Ramanathan and Redi, 2002], which have been in development since the early 70s and have continued growing, thanks to the advances in wireless communications.

A Mobile Ad Hoc Network (MANET) [Mohapatra and Krishnamurthy, 2005] is a collection of wireless mobile devices (nodes), which are able to form a network without the support of any infrastructure or centralized control. MANETs are multihop networks, in which a packet that is sent from a source to its destination must traverse a path formed by two or more hops (nodes); therefore, every node in a MANET has a dual behavior as router and host. MANETs are autonomous networks, they can determine their own configuration parameters and recover in case of failure; they are implemented to set up communications for specialized applications, where there is no preexisting infrastructure or where the available one is not the most adequate for the needs of the operation. Figure **2-1** shows an example of a MANET formed by different wireless devices.

2.1. Features and challenges of MANETs

MANETs distinguish themselves from other networks because they are able to configure themselves autonomously, therefore, there is no centralized control and they can auto recover in case of failure. The topology where MANETs are deployed is dynamic, due to the movement of the nodes[Ortiz and Bobadilla, 2003]. Moreover, the links between the nodes

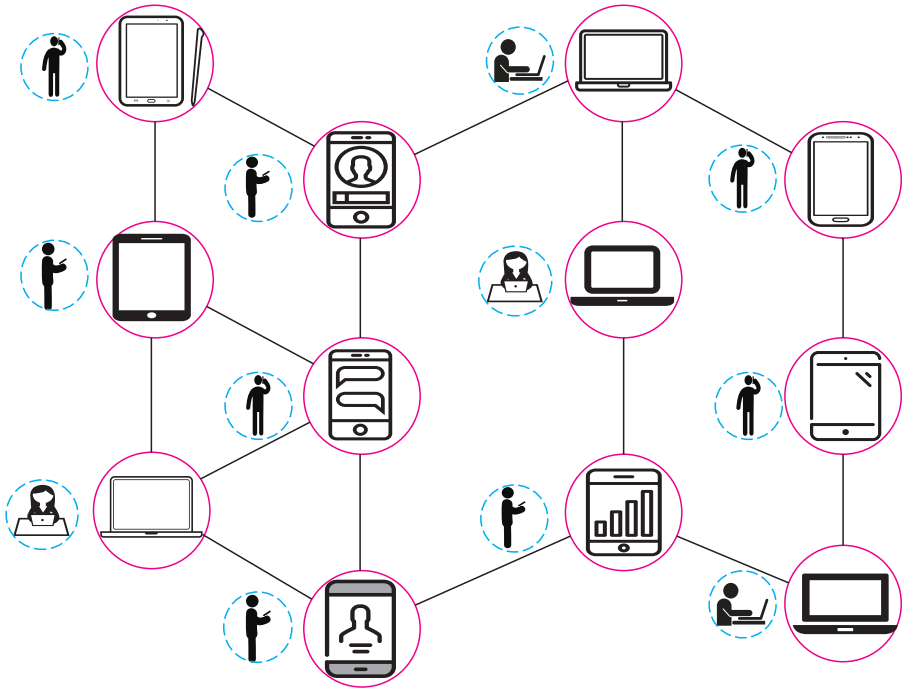


Figure 2-1.: Mobile Ad Hoc Network.

are temporal since they are moving continually, this can cause some instability's. For MANETs, scalability can be a problem, provided that as the network grows, its performance cannot decrease and must maintain acceptable levels of quality for the services offered. Since the nodes of the network do not have a continuous supply of energy and they depend on their batteries, it is necessary for every node to make proper use of its remaining energy. Due to the fact that MANETs are multi-hop networks, in which nodes forward packets to other nodes, sharing access to the wireless channel, security is a main topic, since the network may be vulnerable to attacks.

2.2. Wireless mesh networks and wireless sensor networks

Inside MANETs, two types of them can be found: Wireless Mesh Networks (WMNs)[Akyildiz et al., 2005] and Wireless Sensor Networks (WSNs)[Akyildiz et al., 2002]. These two types of MANETs differ from regular ones, because of some aspects of their operation and hardware specifications. Unlike regular MANETs, in which a node can function as router and host, WMN nodes are classified in mesh routers and mesh nodes: mesh routers have minimal mobility, provide access for regular and mesh nodes, can communicate with other mesh routers, they are in charge of routing, bridging and network functions, and do not have power limitations; on the other hand, mesh nodes can be stationary or mobile and they require efficient use of their energy supply like regular MANET nodes. WSN are conformed by a set of wireless sensor nodes, which are usually deployed in hostile environments and employed for event detection

(e.g. temperature, pressure measure. etc.), these sensors are able to perform some processing over the information obtained and transmit the data over the network, which will allow to the final user understand better the current state of the environment. Unlike MANETs or WMNs nodes, WSN nodes are less expensive than regular wireless mobile devices, smaller and have less hardware characteristics and power consumption; nevertheless, due to their nature of their operation, once a node has exhausted its battery or has been damaged, they may be never retrieved.

2.3. Cooperation in MANETs

Cooperation as such is a strategy employed by a group of entities that work together to achieve a common goal; through this strategy the entities that cooperate benefit thanks to a joined interest. In wireless networks cooperation has many aspects, one of the most important is communication, since it involves a collaborative effort between the devices that conform them. On MANETs cooperation has an additional aspect, a social one; provided its dynamic behavior during its establishment and maintenance which relies completely on the nodes that can decide their level of commitment to the network thus causing some kind of impact on the performance of the network[Fitzek and Katz, 2006].

Since MANETs are networks with a very particular fashion of operation, it is necessary for all of the nodes to cooperate altruistically in order to make up for the absence of an infrastructure[Chlamtac et al., 2003, Hoebeke J., 2004]; however, provided that the nodes may not be homogeneous and have hardware limitations, if cooperation arises, each node would have to use its limited resources to maintain the operation of the network, consequently, cooperation does not bring any direct benefit to the nodes, and because of this, selfish behaviors can emerge. A Selfish node will only cooperate if it receives direct benefit from cooperation, in addition, a selfish node will expect from the other nodes to cooperate with it, to gain benefit without using its own resources[Misra, Sudip, Isaac Woungang and Misra., 2009].

Considering that, in MANETs the main goal is to maintain the communication and the services that are being executed. Despite of the changes that may occur, several authors have proposed different methods to stimulate cooperation and avoid selfish behaviors. To stimulate cooperation [Zhong et al., 2003] has proposed a system of payments, in which nodes that cooperate are rewarded with tokens that will allow them to access for services offered in the network when needed. Another method that has been proposed uses reputation mechanisms[Rebahi et al., 2005] in which the reputation of the nodes that cooperate increases, and for those who do not do it, decreases and eventually are excluded from the network.

With this in mind, it is easy to deduce that MANETs in nature should be altruistic and the nodes must find a way to cooperate with each other, under any kind of circumstance[Conti et al., 2004].

2.4. Contents management in MANETs

On one side, nodes have a dual behavior as router and host; on the other side, when it comes to multimedia content that is streamed, shared or stored in the network, nodes can behave as servers and clients as well[Yin and Cao, 2004]. Managing contents in MANETs also has its own challenges, facing many issues that are caused by the nature of MANETs[Rubinstein et al., 2006] as well. Mobility is one of the key features of MANETs, since it can increase the coverage area of the network; however, when it comes to content management it can pose one of the biggest issues[Chlamtac et al., 2003], because it can cause partitions, segmenting the network and decreasing the availability of the data, for instance, at some time a node may move far away from the network to a spot where it is unreachable, this node can have important information, and once this node has left the network, it won't be available for the other nodes to access. Energy is a topic of attention as well, if nodes start exhausting their battery, this can be another reason for them to leave the network, the more nodes leave, the more sparse it becomes, decreasing the availability of data.

Replication and allocation of data inside the network are techniques employed to ensure data availability provided the advantages they can bring[Hara, 2001]; for example: they can decreases the number of hops that a packet must traverse to reach its final destination, helping the network save its energy resources; nonetheless nodes may have poor resources, may not be heterogeneous and under some circumstances they may be leaving the network, that is why it is necessary to allocate hot data in such way, that it is stored in nodes with enough hardware support to ensure its availability regardless of the situations that may be happening inside the network.

In figure **2-2**, it is shown all the issues that can affect the availability of the data due to partitions in the network.

2.4.1. Methodologies to manage contents in MANETs

Considering the way MANETs work, it is very clear that they are prone to partitions by different causes and that the data accessibility is lower compared to conventional networks, then it is essential to develop new techniques that can help to solve these inconvenients, among those techniques, the following can be found:

Quorum System

Distributed systems such as MANETs can be conformed by a system of quorums [Friedman et al., 2010], this kind of system is very useful to achieve consensus. Through quorums its to possible to ensure that the nodes which belong to a sub system intersect each other, allowing the system to maintain consistency internally.

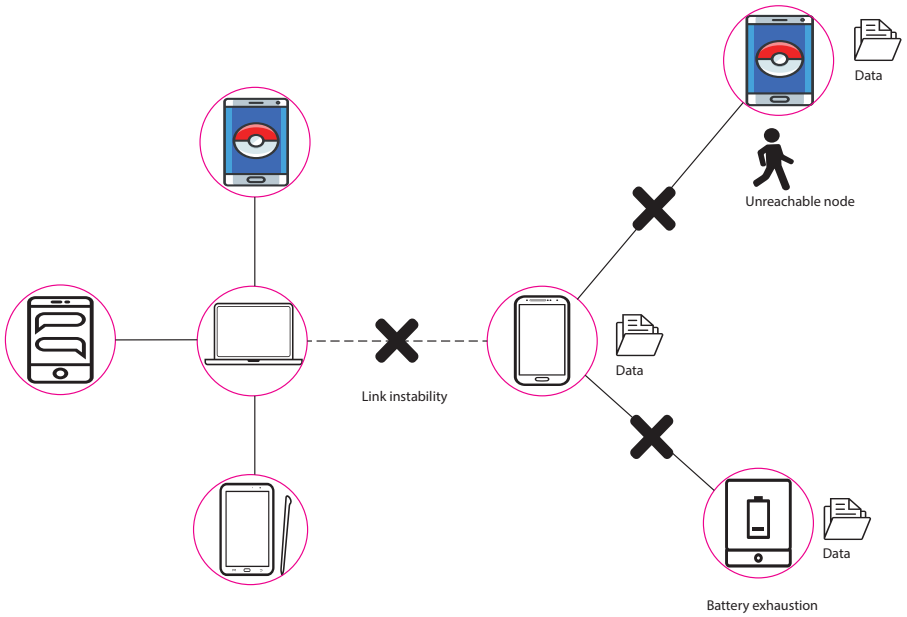


Figure 2-2.: Network partition causes inside a MANET.

Lemma 1. (Set of systems) *A set of systems S over a universe U, is a set of subsets of U.*

Lemma 2. (Quorum system)) *A set of quorums (Quorum System) Q over a universe U is a set of systems over U in such way that for any $Q_1, Q_2 \in Q$, and $Q_1 \cap Q_2 \neq 0$.*

Quorum System is a technique widely used to replicate contents in MANETs, more utilization cases and its modifications, proposed by different authors can be found on [Mannes et al., 2012, Kanzaki et al., 2008, Karumanchi et al., 1999, Gilbert et al., 2010, Friedman et al., 2010, Sawai et al., 2006, Padmanabhan et al., 2008].

A replica allocation method adapting to topology

In this paper[Hayashi et al., 2005], they propose a proactive method to allocate replicas inside a neighborhood of nodes that are N hops away; each node stores its neighbor's information, and relocates the replicas as soon as it detects abnormal situations that may cause partitions in the network. The method uses a series of messages to keep the information table of node neighbors updated every time that there are changes, for instance: when a node leaves due to an anomaly, if an anomaly is detected or if it's necessary to relocate the information. To replicate and allocate the data, the method considers the access frequency of the contents stored, the method assumes that the contents are always consistent and have the same size. When unusual behavior that may disrupt the operation of the neighborhood is detected, if a node detects that there is only one route towards a N-hop neighbor or if after launching a loop it does not return to the source, the contents stored in this abnormal node are requested and relocated according to the frequency in which contents are accessed and the capabilities

of the nodes inside the N-hop neighborhood. As can be seen, by detecting proactively unregular node behavior, it is possible to predict when a node may leave the network, hence, allowing content relocation.

Data replication considering power consumption in Ad Hoc networks

In [Shinohara et al., 2007] 4 methods to replicate and allocate data within a MANET are proposed, each method aims at managing contents in such way they are not gathered in a single location and that the whole network energy consumption decreases. To achieve this, each method takes into consideration the access frequencies, the number of the replicas and the remaining battery power to calculate a value, and it makes a decision about what to do with the contents.

The first of these methods is called Expected Access (EA), in this method each node creates a replica of the frequently accessed contents by itself and its neighbor peers that are stored by a small group of nodes; this method balances energy consumption, since contents that are highly accessed are replicated in a great number of nodes. The second method is called Weighted-EA (WEA), like the previous method, it considers the access frequencies of a node and its neighbors to a set of contents, each node calculates the access frequencies of itself and its peers to the contents it holds; however, unlike the previous method a priority value is added; this method also balances energy consumption; but just per node, because only contents accessed by node will be replicated and not by its neighbors. The third method is called WEA-Battery (WEA-B), it pretty similar to the WEA method, but it considers the remaining energy of the node. The last method is called WEA-Hop (WEA-H), it is similar to the second method, but this one considers the path length between nodes, since a long path will result in more energy consumption.

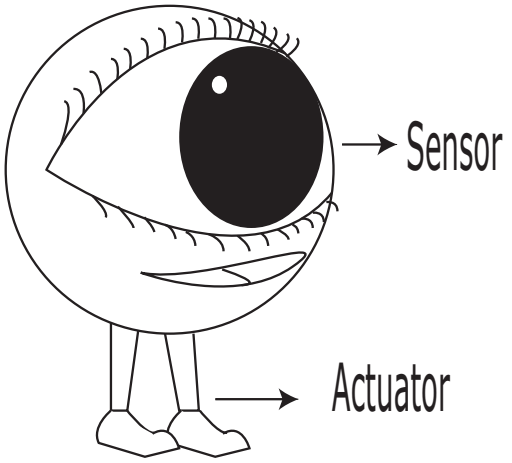
3. Bacteria agent population inside a MANET

A MANET, a multi agent system (MAS) and a population of bacteria, at first may look as three unrelated topics; however, they share similarities and patterns that emerge once the parts of each system start interacting. In this chapter, these similarities will be reviewed and a bio inspired algorithm from Quorum Sensing will be proposed. Bio-inspiration[Babaoglu et al., 2006] is not new on MANETS, Bio-inspired multi agent systems have been employed before, the agents for these cases have been endowed with behaviors similar to insects such as ants[Martins et al., 2010], bees[Wedde and Farooq, 2005] and termites[Hoolimath et al., 2012], provided that, these type of behaviors are adaptable and highly scalable.

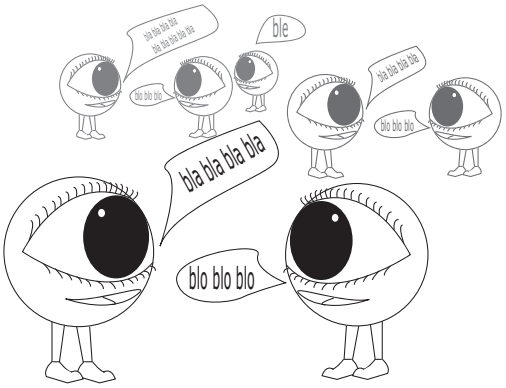
3.1. Agents and their environment

An agent is an entity with a certain degree of autonomy that exists inside an environment and reacts according to the feedback received from it. An agent can perceive its environment through sensors and act upon it through effectors or actuators[Russell and Norvig, 2010]; by itself an agent is not capable of performing huge or complicated tasks; nevertheless, by interacting with other agents, they can build a system (MAS), allowing them to solve problems or optimize different processes. Figure **3-1**a shows the structure of an agent.

For MAS to exist and be able to interact, an environment must exist, with all the necessary conditions[Panait and Luke, 2005, Weyns et al., 2005]; the environment is an independent entity, whose changes do not depend on the interaction of the MAS. Communication is a key element for the agents to coordinate themselves; an agent can make better decisions if it receives enough information from other agents and the environment. The environment can be used as an indirect medium of communication, as agents can alter states of the environments, and other agents can sense these modifications as messages, in a similar manner to the pheromones employed by some insects. Figure **3-1**b, shows a MAS interacting with each other and their environment.



(a) Agent structure.



(b) MAS interaction over the environment.

Figure 3-1.: Agents.

3.2. Bacteria and quorum sensing

A bacterium is a single cell microorganism that isolates itself from other bacteria thanks to a membrane, they lack a defined nucleus and instead they have a rounded chromosome[Madigan et al., 201 Bacteria reproduce through binary fission, once a cell has grown; it splits itself into two identical cells. Bacteria are diverse, they have adapted themselves to habit in different types of environments; they can be found in places without oxygen, where the environment is too hot, acid, cold, etc. Bacteria organize themselves in populations, where they cohabit with bacteria of different species, with whom they share the environment; the interaction between bacteria of different species can be beneficial, neutral or harmful.

The environment is important for the development of the bacteria population, since it provides all the necessary nutrients to metabolize and reproduce. Even though bacteria, can survive in hostile environments, this does not imply that all bacteria can survive everywhere. An environment can benefit one population but be very harmful to another[Madigan et al., 2013]. In figure[1] **3-2** there is a bacteria population interacting in their environment.

Bacteria are able to communicate with bacteria of the same or different species, utilizing chemical signals, which are synthesized and secreted by them[Federle and Bassler, 2003]. With the information contained in these signals, bacteria can synchronize their population and carry out large scale collective actions, which are effective when they are performed by the whole population, but ineffective when performed by a single bacterium.

Communication between bacteria is achieved through a mechanism known as Quorum Sen-

[1]free image taken from: `https://pixabay.com/es/bacterias-especies-bacterianas-106583/`

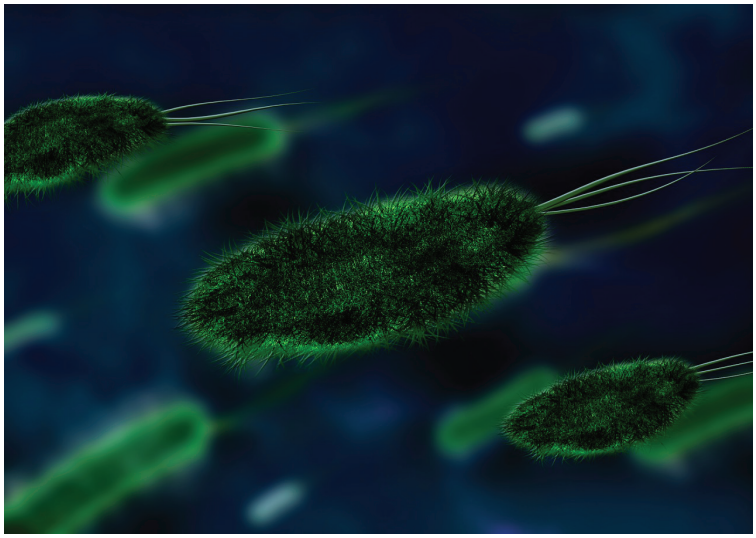
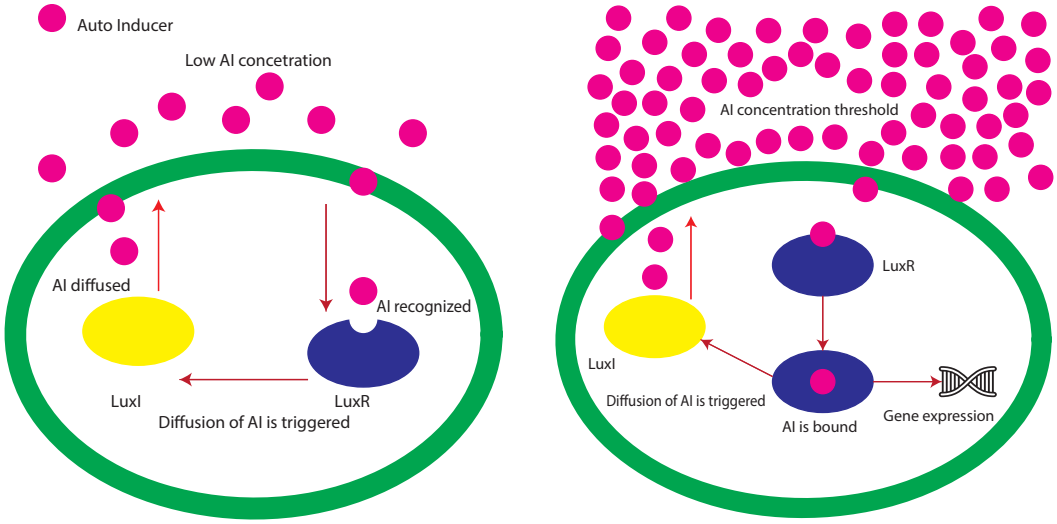


Figure 3-2.: Bacteria and their environment.

sing (QS)[Jayaraman and Wood, 2008, Waters and Bassler, 2005, Williams et al., 2007], which enables the control of various bacteria phenotype such as motility, antibiotic biosynthesis, bioluminescence, biofilm formation, among others. As mentioned before, bacteria utilize chemical signals (molecules) which are called Auto Inducers (AI). The AI signals allow bacteria to monitor their environment, count their population and communicate with the others. QS was the first described for Vibrio Fischeri (V. Fischeri)[Nealson et al., 1970, Eberhard et al., 1981], a species of bacteria, which can be found living on the skin of squids. V. Fischeri utilizes QS to produce light (bioluminescence), the relationship between the population of V. Fischeri and the squid is beneficial for both parties, the light produced by the V. Fischeri helps the squid to avoid predators, attract preys among others; V. Fischeri gains an environment full of nutrients on the skin of the squid. This discovery has helped to establish the paradigm of QS and to describe the circuit employed for QS communication.

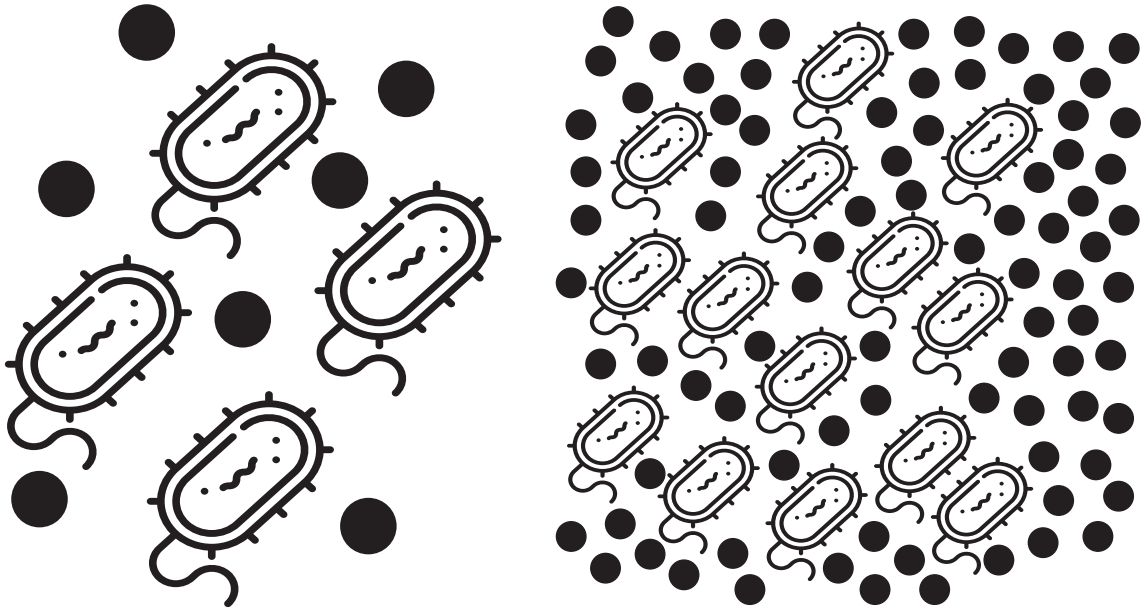
The QS circuit for communication is shown in figure **3-3**, this circuit employs two regulatory enzymes (LuxI and LuXR), which synthesize and recognize the AI signals. LuxI synthesizes the AI, which is diffused inside and outside of the cell, as shown in figure **3-4**a, at low cell densities the amount of AI (the small dots in the figures) within the environment is low; once the cell density starts increasing so does the concentration of the AI, as shown in figure **3-4**b. When the concentration of the AI reaches a minimum threshold level, the AI is bound to the LuxR enzyme as shown in figure **3-3**b, the gene expression is activated (bioluminescence in the case of V. Fischeri) and thus QS is attained.

The AI[Bassler, 2002] plays an important role in QS, not only is the signal that starts the process but, it is also the signal that triggers the emission of more AI, this means that when AI is recognized by the LuxR enzyme, another AI will be diffused as shown in figure **3-3**a,



(a) Low AI concentration. (b) High AI concentration.

Figure 3-3.: Quorum Sensing circuit.



(a) Low population density. (b) High population density.

Figure 3-4.: Population density and AI concentration levels.

this creates a positive loop. The final behavior of the population depends on the concentration of the AI sensed in the environment. The concentration threshold then becomes an important factor to achieve QS, since the AI concentration must reach a certain threshold level, in which all the bacteria change their state at once and carry out large-scale population traits. In order to achieve this threshold, once the AI has been recognized, it triggers the diffusion of more AI, when the threshold concentration is met; the rate of emission of AI rises dramatically. QS is very sensible to changes in the concentration, provided that the threshold concentration determines if QS is performed or not and at the same time the concentration depends on the population density.

QS is not exclusively performed by gram negative bacteria; but also by gram positive bacteria as well, further explanation of the process for these bacteria can be found on [Kleerebezem et al., 1997, Gobbetti et al., 2007].

3.3. Agents as bacteria within MANETs

An agent within a MANET for this research, can be envisioned as an entity endowed with abilities similar to bacteria, their interaction with other agents would allow them to build a population and their communication would obey the QS principles.

One of the many traits that are result from the QS are the biofilms[Li and Tian, 2012, Parsek and Greenberg, 2005], which are structures where bacteria cohabit and develop specific roles within their population, with this in mind, it's possible to make an analogy between bacteria within a bio film and agents inside a MANET; using bio-inspiration, some characteristics of the QS can be applied to a MAS over a MANET. Provided that bacteria release molecules as a mean of communication, receiving constant feedback from other bacteria through the molecules sensed over their immediate environment, they utilize QS as an indirect mean of communication for population wide synchronization. This is a communication strategy, which once understood, can be useful for MAS[Amaris et al., 2015].

QS relies on population density and the amount of AI in the environment. This large-scale behavior emerges from local interactions, a population of bacteria can be conformed by millions of cells, and it is not possible that a single bacterium has all the information about the population, instead it must rely on what it senses from its close environment and bacteria neighbors; according to what it senses, it releases more AI signals, this is a loop, continually executed at a local level until the threshold of AI is achieved. In a similar way an agent doesn't have the whole information about its environment; but it receives constant feedback from its peers by sensing the environment, with these information the agents can coordinate themselves and be able to reach a state of group behavior similar to QS.

Communication for this case is the first step towards cooperation, it involves a signaling system with enough information, to be able to be recognized at a local level and which can trigger previously defined actions; too much information can introduce noise and delays to the communication process but, little information can block it; the information generated by the signaling system needs to be tuned to intermediate point[Amaris et al., 2015]. The second step is coordination, this part involves the system to recognize the signals and act according to the information contained in them, it also involves negotiation and information exchange processes, carried out until a criteria or a set of criteria are met. Finally, in cooperation step all of the parts that have agreed in the previous step perform a collective action. Figure **3-5** shows a diagram of the series of steps explained and the analogies between the agents and bacteria.

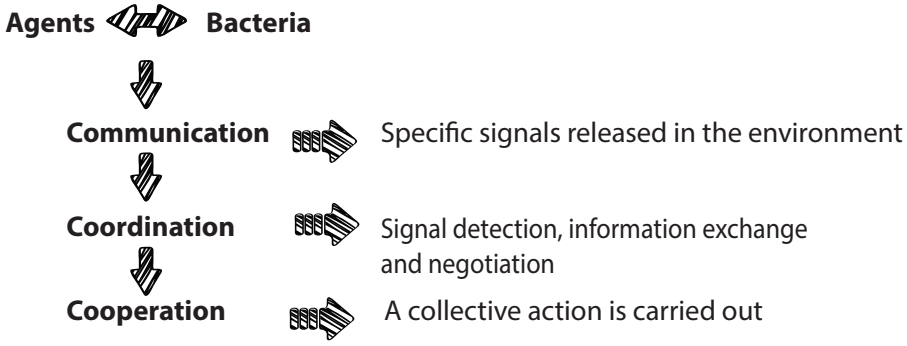


Figure 3-5.: Associations between agents and bacteria.

3.4. Algorithm for agent communication inspired by Gram negative bacteria Quorum Sensing

As is seen in the development of this chapter, QS has a lot of advantages. For this part of the research is the source of inspiration for an algorithm for multi-agent communication within a MANET, this algorithm follows the QS principles of communication to help a population of b-gents synchronize themselves and achieve collective group behavior.

To implement this algorithm it's important to consider:

1. Define a $n - bits$ chromosome.
   - Set 2 bits for the LuxI and LuxR enzymes.
   - Define m bits for the node index which will work as the agent ID.
   - Define 1 bit to check the QS status of a node.
   - Define 1 bit for content storage.

- Define l bits for the file index.
- Define p bits for the amount of molecules that an agent can release.

2. Define a mutation probability for the agent.
3. Define a cloning probability for the agent.
4. Set the number of hops for the agent to move.
5. Define a threshold and a total capacity for molecules for all the nodes.

One of the goals of the chromosome is to build a information table based on what the agent has sensed to help other agents gain more knowledge about the network.

Mutation is only carried out over the LuxI and LuxR bits, by randomly selecting one and inverting its value (0 to 1 or vice versa). After the mutation, it is essential to contemplate the next items:

1. An agent with only the LuxI enzyme, will be constantly releasing signals each time it arrives a node; nevertheless, it is not able to induce the QS state on a node.
2. An agent with only the LuxR enzyme will sense, induce nodes to the QS state and update its status; nonetheless, it will not be able to release signals in the environment.
3. A agent with both LuxI and LuxR enzyme is able to participate in all the processes associated to the services offered by the network.
4. In case that an agent without the LuxI and LuxR enzymes is produced, its chromosome will be set to its default value, having both the LuxI and LuxR molecules active.

3.5. Penalization

The algorithm will exercise penalization over the concentration of molecules in the node if the next situations occur:

- To predict when a node may leave the network, the agent will use the routing information of the once it arrives a node and checks its routing table as suggested by [Hayashi et al., 2005], when a node has less than 2 neighbors, its concentration will be penalized and if it holds any content and has QS neighbors, the content will be immediately relocated to its closest peer immediately.
- When a node does not fulfil the requirements for the application.

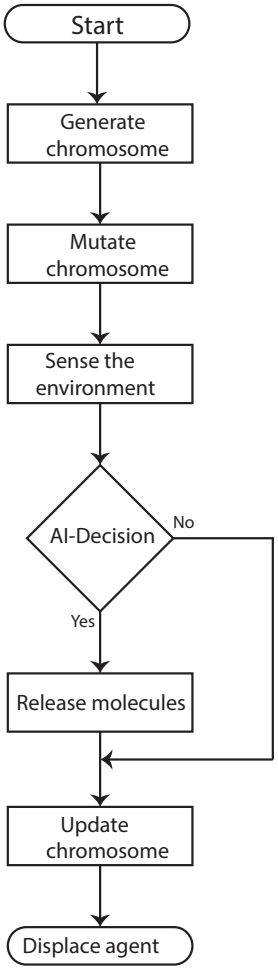


Figure 3-6.: Creating the agent.

3.5.1. Creating the agent

The diagram on the figure **3-6** shows the first part of the algorithm, which follows the next steps:

1. The application updates all of the variables of interest (disk space, energy, routing and neighbor's information) for the agent to sense once it arrives a node.

2. Define a mutation percentage m, and proceed to generate a random number w between 0 and 1 if $m \leq w$, mutate the chromosome, otherwise, continue to the next step.

3.5.2. Multi-agent QS based communication

The first part of the algorithm which involves the creation of the agent is shown in figure **3-6**.

In figure **3-7**, there is the process that an agent must follow once it arrives a node. This strategy adds additional information to the environment, this information is enough to help the agents while moving across the network to ensure data availability. Note that the agents are also used as a mean of indirect communication by updating their chromosome according to the node they find themselves in, and by supplying that information to other agents when they update the neighbours information table.

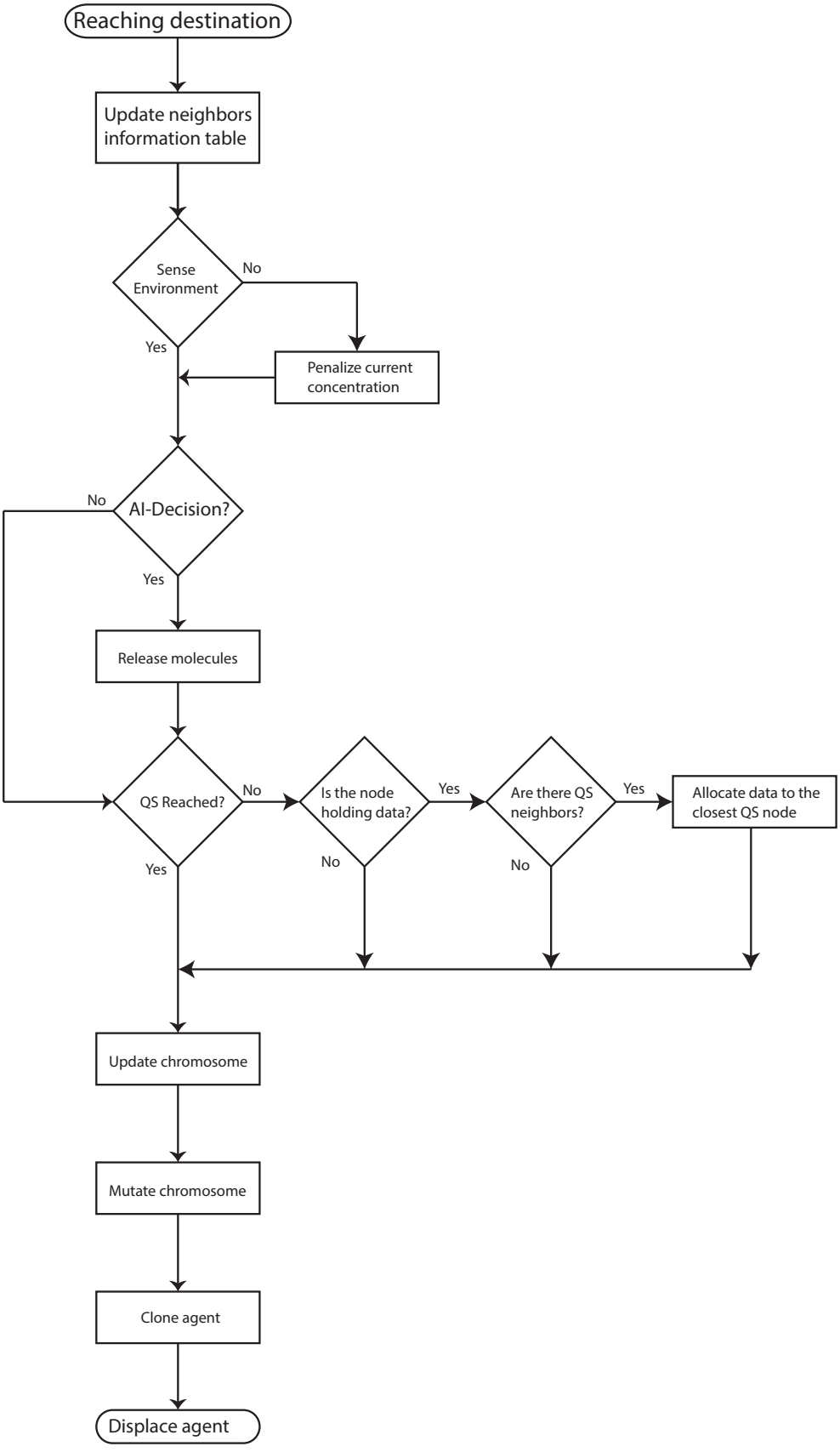


Figure 3-7.: Bacterial agent QS based communication.

4. Microeconomics based multi-agent decision making

4.1. Decision making

Bacteria are never alone, a single bacterium is not able to perform an action that can cause any impact within the population; nevertheless, considering their limited cognitive abilities, once they have organized themselves in populations, they can coordinate themselves and thus be able to perform collective decisions, based on the signals sensed from the environment. Decisions are made under any kind of circumstance, by making the best decision, it is always possible to perform a given task in the best possible way; however, in situations where there are several criteria which may conflict with each other, a single solution won't be reached; instead, a collection of solutions that satisfy the multiple conflicting criteria[Ehrgott and Gandibleux, 2002].

For bacteria, while interacting over the environment, in order to make decisions, each bacterium monitors its close environment, they rely on the "social information"[1], which is not fully detailed; but, allows the bacteria to obtain a good approximation. Completely detailed information, under any circumstance, not only for bacteria but, for other types of populations, brings more accuracy to the decision making process; however, detailed information is very difficult to obtain, incurs in more costs and takes longer processing. Therefore, if its necessary to make a quick decision because the environment can change unexpectedly so, accuracy is not relevant and can delay the whole process, instead it is better to settle with less information, which can make the decision process adaptive and flexible[Ross-Gillespie and Kümmerli, 2014].

Due to QS, bacteria are releasing molecules all the time into the environment, which can lead to a positive feedback, because the molecules are accumulating, it could speed up the decision making process; nevertheless, under situations when is necessary to choose from a discrete set of options, positive feedback can favor the first good options found, discarding possible better options which could remain undiscovered. In order to prevent a situation like this, it is necessary to implement a mechanism to counter balance positive feedback

[1]Signals produced by other individuals[Ross-Gillespie and Kümmerli, 2014]

and adjust the global response, performing negative feedback can achieve this. By using both positive and negative feedback, modulation of the collective decision process can be accomplished[Ross-Gillespie and Kümmerli, 2014].

A quorum can be understood in different ways according to the context in which the concept is employed[Sumpter and Pratt, 2009], for instance, in some social meetings, a quorum is the minimum number of participants necessary before any decision is made; in some groups of animals, is the minimum group of members in favor for a particular choice. However, neither of these definitions of quorum applies for QS. As mentioned before QS is associated with the molecules released into the environment, the accumulation of these molecules works as an index of population; however each bacterium produces different amounts of molecules, which means that the relation between the population and molecules is not lineal; therefore, in the bacteria context, quorum is more associated with the AI concentration threshold in the environment, after this threshold is surpassed, a response is effected.

4.1.1. Decision-making mechanism based on microeconomics for agents

Economics is the science that studies how scarce resources are allocated; a resource is said to be scarce if its supply is limited. In economics there are two branches: macroeconomics and microeconomics. Macroeconomics studies the behavior and performance of an economy as a whole, for example, how the economy of a country works. Instead, microeconomics, studies the behavior of individual economic agents (consumer and producer), the decisions they make and how their interaction builds a market[Mankiw, 2006, Besanko and Braeutigam, , Krugman and Wells, 2015, Pindyck and Rubinfeld, 2009, Nicholson and Snyder, 2008].

As other sciences, economics uses models as well, these models are used as tools to simplify the reality and are based on assumptions; also, they are mathematical representations of economic theories. Most of the economic models have as starting point an assumption about economic agents going after some goal rationally. Microeconomics also makes use of different economic theories to study the interaction of the economic agents. One theory in particular is the consumer's theory, that describes how they assign their income to a collection of goods and services expecting to maximize their welfare. To understand better the behavior of the consumers, first its important to understand 3 topics from the consumer's theory: the first one is about their preferences, finding a way to describe how consumers prefer one good to another; the second one, is about budget restrictions, consumers have a limited income, which imposes a limit to the amount of goods they can purchase finally, the third one is about their preference choices and their limited income, they buy a combination of goods, expecting to increase their satisfaction.

In a regular market, goods are grouped in baskets, which contain specific amounts of one or more goods; the consumer's theory starts with three assumptions to study the preferences of a consumer while she compares the goods of the baskets:

1. Completeness: preferences are complete, consumers can compare and order their available baskets, for instances if there are two baskets A and B, a consumer will prefer A to B, B to A or will be indifferent towards the 2 baskets, which means that both baskets give her the same level of utility.

2. Transitivity: Preferences are transitive, for example if a consumer prefers A to B and B to C, she prefers A to C.

3. The more the better: consumers are never satisfied with what they have and will always prefer higher amounts of goods.

To a better understanding of consumers preferences, it's also necessary to employ another tool: the indifference curves, these display the combination of different goods that bring the consumer the same level of utility. As can be seen in figure **4-1** the slope of the curve is negative, the consumer is willing to give up a certain amount of good y as long as she is compensated with aggregated amounts of x while maintaining the same level of utility; the slope becomes greater or less negative when x increases which means that a consumer will be less willing to give up amounts of y to gain more amounts of x. The negative slope on a given point over the indifference curve is called marginal rate of substitution (MRS) and it represents the amounts of a good a consumer is willing to give up to gain more units of another, as long as the new good bring her the same level of utility(k).

Provided that the utility function for two goods of a consumer is given by $u(x, y)$, then the MRS is given by:

$$dU = \frac{\partial U}{\partial x}dx + \frac{\partial U}{\partial y}dy \tag{4-1}$$

Considering that the value of the utility is a constant (k) along the curve, then $dU = 0$, which implies:

$$MRS = -\frac{\mathrm{d}y}{\mathrm{d}x}(U = k) = \frac{\frac{\partial U}{\partial x}}{\frac{\partial U}{\partial y}} \tag{4-2}$$

Indifference curves can be obtained by employing utility functions, which assign a numerical value (utility) to a basket of goods, one utility function is the Cobb-Douglas given by equation 4-3:

$$U(x, y) = x^{\alpha}y^{\beta} \tag{4-3}$$

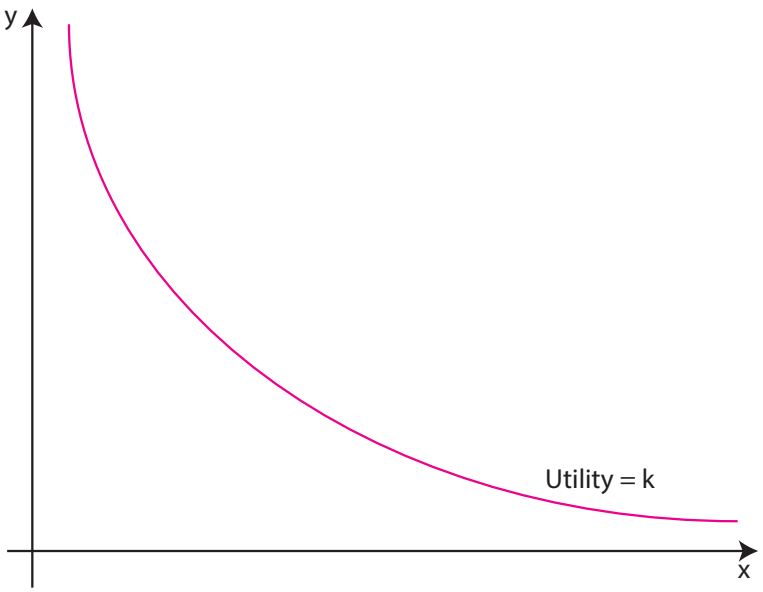


Figure 4-1.: Indifference curve

In the Cobb-Douglas utility function α and β represent the preference given to the goods x and y respectively, these two values are normalized in such way that $\alpha + \beta = 1$.

A consumer will always try to maximize her utility given her fixed and limited income to spend, she expects to acquire quantities of goods that exhaust her income and find a point in which the MRS is equal to the rate in which one good can be traded one for the other[2]. More goods will bring more utility; nevertheless, budget constraints forbid her to buy more goods and not spending all her income would fail to maximize her expected utility.

Figure **4-2** shows the curve for a budget constrain for two goods x and y given by equation 4-4, where P_x and P_y are the prices for goods x and y respectively and several indifference curves.

$$I(x, y) = P_x x + P_y y \tag{4-4}$$

Provided that is of interest to find the amount of goods that maximize the utility given the limited income, by imposing the budget constraint over an indifference map[3] it is possible to find a point that fulfills all the requirements, by analyzing figure **4-2** it is possible to infer that:

- Point A on the curve U_1 leaves income unspent, which means that a consumer can gain more utility by spending a greater portion of her income on more goods.

[2]the slope of the budget constatint equation
[3]An indifference map shows a collection of indifference curves

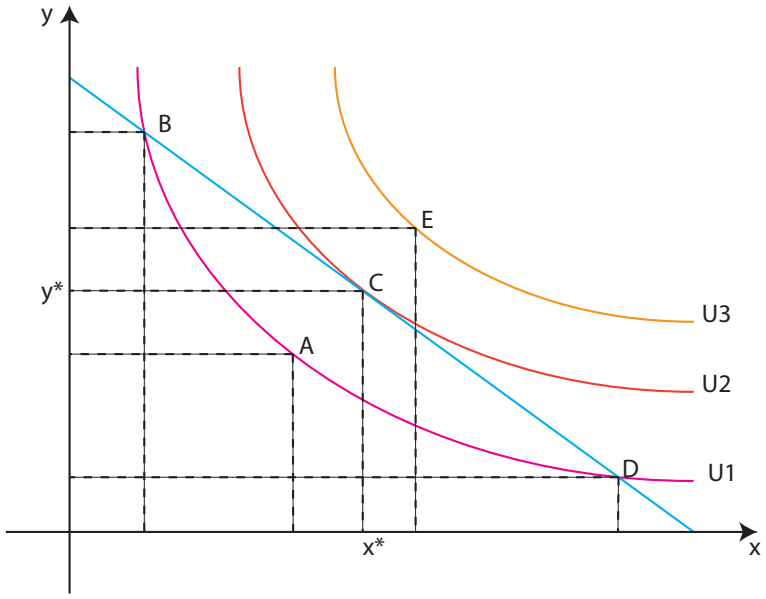


Figure 4-2.: Optimal choice given a budget constraint.

- Points B and D, which are on the same indifference curve as point A, bring the same level of utility to the consumer; nevertheless neither of this points maximizes the utility.
- Point C on the curve U_2 is on a higher position and to the right, which means that the utility supplied by indifference curve U_2 is greater than U_1. Also this is the point of tangency between the budget restriction and the indifference curve, and this is the point in which $x*$ and $y*$, are the amounts of goods that fulfill the restrictions.
- Finally point E, which is on curve U_3 and is on a higher position than any point on the map of indifference curves, brings more utility to the consumer; nonetheless, due to the fact that neither point of curve U_3 is tangent to the budget restriction, the prices are too high and the income of the consumer is not enough to acquire goods over this curve.

Point C which is the tangent point between the budget constraint and the indifference curve, is the optimun and the point where the slope of the budget constraint is equal to the slope of the indifference curve and equal to the MRS, as is shown by equation 4-5.

$$-\frac{P_x}{P_y} = -\frac{\mathrm{d}y}{\mathrm{d}x}(U(x,y) = K) = MRS \tag{4-5}$$

4.1.2. Multi-agent decision-making mechanism

While interacting inside the MANET and using the algorithm from the previous chapter, the agents must sense the network and make decisions. Since MANETs are networks with scarce resources, it is possible to apply microeconomic concepts along with consumer's theory to

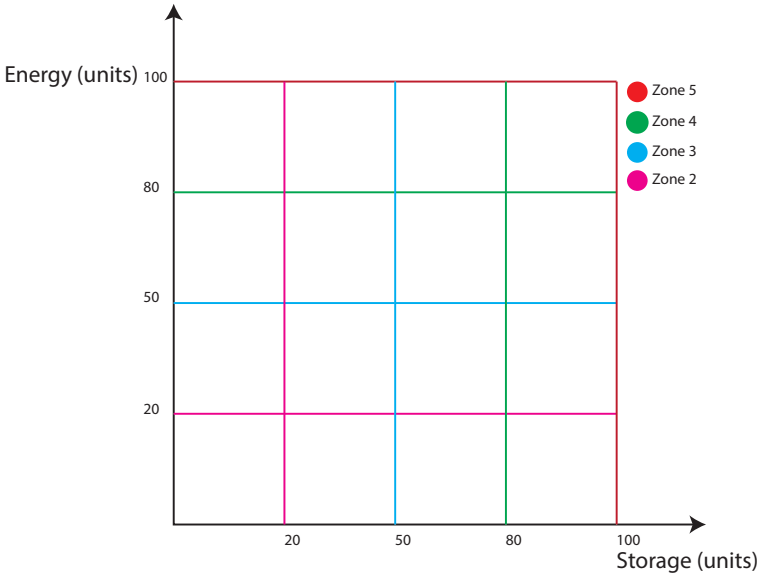


Figure 4-3.: Threshold zones.

give the agents all the necessary tools to make decisions.

In order to make decisions, while sensing the nodes, agents will only consider the remaining energy and disk space. Provided that both resources are insufficient and there is no chance that they will be upgraded during network operation, in order for agents to ensure data availability they must choose the most suitable nodes; to achieve this, agents will use a decision making process based on utility functions and the Paretto dominance[Deb et al., 2002]. The process works in this way:

1. Determine the threshold levels of current remaining energy and disk space: The amounts of units of x (disk space) and y (energy) are ranked according to threshold zones, each zone is defined as follows (figure **4-3**):

   - Zone 2 for values of x and y between $0 < x, y \leq 20$.
   - Zone 3 for values of x and y between $20 < x, y \leq 50$.
   - Zone 4 for values of x and y between $50 < x, y \leq 80$.
   - Zone 5 for values of x and y between $80 < x, y \leq 100$.

   In order to increase the utility that each agent obtain, α and β are calculated using the "m" (index zone for disk units) and "n" (index zone for energy units) parameters as follows:

   - If $m = n$, both $\alpha = \beta = 0.5$.
   - If $m > n$, then $\alpha = \frac{m}{m+n}$ and $\beta = 1 - \alpha$.

- If $n > m$, then $\beta = \frac{n}{m+n}$ and $\alpha = 1 - \beta$

2. the agents solve the next optimization problem:
   Maximize:

$$U(x, y) = \alpha \cdot log_m(x) + \beta \cdot log_n(y) \tag{4-6}$$

Where α and β are the respective preferences for x and y and are normalized in such way that $\alpha + \beta = 1$; m and n are the zone indexes obtained in the previous step for x and y respectively.
Restricted to:

$$I(x, y) = P_x x + P_y y \tag{4-7}$$

Where $I(x, y)$ is the total amount of molecules (AI) an agent can release in a node, and P_x and P_y are the respective prices of x and y. For this research the agents, are price takers and each node can set the prices of the two goods it offers.

Before finding $x*$ and $y*$ it is important to give enough support that the equation 4-6 is in fact a utility function, as seen previously the indifference curves are convex[Walter Nicholson and Christopher Snyder, 2008], this can be proved with the MRS as follows:

a) Calculate the partial derivatives for x and y.

$$\frac{\partial U(x, y)}{\partial x} = \frac{\alpha}{ln(m) \cdot x} \tag{4-8}$$

$$\frac{\partial U(x, y)}{\partial y} = \frac{\beta}{ln(n) \cdot y} \tag{4-9}$$

b) Calculate the MRS.

$$MRS = \frac{\frac{\partial U(x,y)}{\partial x}}{\frac{\partial U(x,y)}{\partial y}} = \frac{\alpha \cdot ln(n) \cdot y}{\beta \cdot ln(m) \cdot x} \tag{4-10}$$

It is evident that the MRS decreases as x increases and y decreases, therefore the indifference from equation 4-6 are convex. To find $x*$ and $y*$, the agents use the lagrangian

multiplier method[Walter Nicholson and Christopher Snyder, 2008] obtaining as result for $x*$:

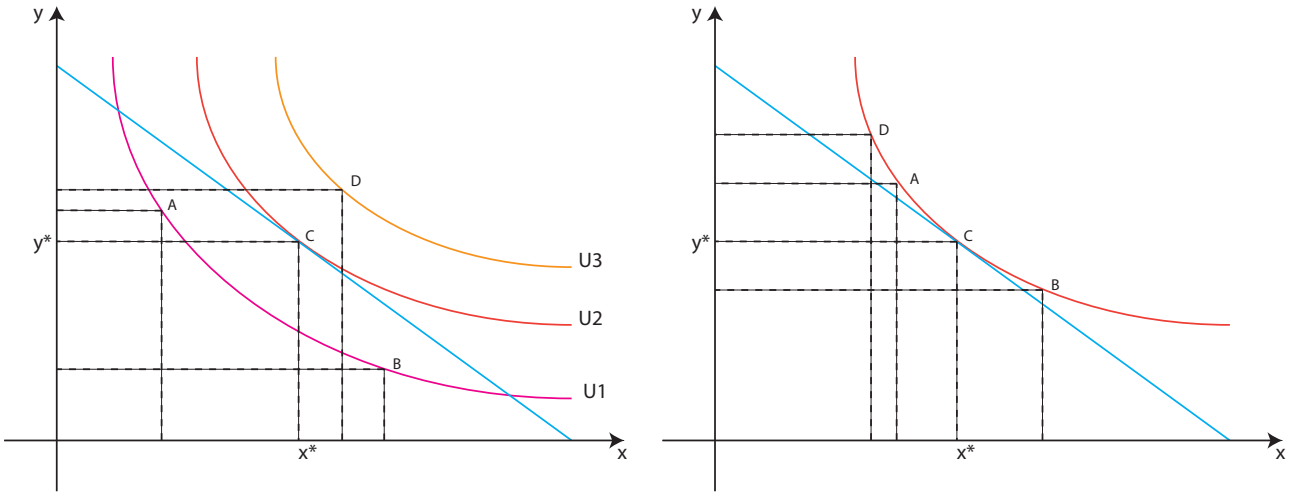
$$x* = \frac{\alpha \cdot log_m(e) \cdot I(x,y) \cdot P_y}{P_x(\beta \cdot log_n(e) + \alpha \cdot log_m(e))} \tag{4-11}$$

and for $y*$:

$$y* = \frac{\beta \cdot log_n(e) \cdot I(x,y)}{P_y(\beta \cdot log_n(e) + \alpha \cdot log_m(e))} \tag{4-12}$$

3. According to the consumer's theory $x*$ and $y*$ are the amounts that maximize the consumer's utility and exhaust her budget. Since $x*$ and $y*$ are on an indifference curve, now it is of interest to determine if the current amount of available units of x and y in the node are on the same curve as $x*$ and $y*$. To find this out, the agents will use Pareto dominance, on the one hand points over the same indifference curve bring the same level of utility to the consumer, and on the other hand, points on the same Pareto front satisfy the same restrictions of a problem and neither point is better than the other; with this in mind, it is possible to make an analogy between a Pareto front and the indifference curve. Applying the Pareto dominance on $x*$ and $y*$, and the current amount of x and y available in the node, the agent can make a decision with two different outcomes:

   - If $x*$ and $y*$ dominate the current amounts of x and y, this means that $x*$ and $y*$ are on higher indifference curve. According to the consumer's theory, the agent will leave a portion of its molecules unused, which won't maximize its utility. On the contrary, if x and y dominate $x*$ and $y*$, even though the current amount of available units will give the agent more utility, due to molecule restrictions, the agent does not have enough molecules to release in this node, since the current units are on a higher indifference curve. If either of these situations is presented to the agent, it will reject the node and move to another node, for example in figure **4-4**a, point C on U_2 dominates any point on curve U_1; nevertheless, it is dominated by any point on curve U_3.
   - If $x*$ and $y*$ do not dominate the current amounts of x and y, and x and y do not dominate $x*$ and $y*$ either, both points are on the same Pareto front and since neither of them is better than the other, they are on the same indifference curve as well and will give the same level of utility to the agent, it will accept the node and release all of its molecules in the node. For instance in figure **4-4**b, point C is on the indifference curve as points A, B and D, which means that neither of them dominate the others and all of them bring the same level of satisfaction.



(a) Dominated points (b) Non dominated points

Figure 4-4.: Dominance criteria for multi-agent decision making.

5. File segmentation methodology for B.S.A.A application

The main goal of this dissertation is to employ a MAS to manage contents in a MANET, an algorithm based on QS for multi-agent communication and a microeconomics based decision mechanism have been proposed in this research. The contents for the application will be segmented into pieces and that is how they will be stored in the network.

As mentioned in chapter one, MANETs are not stable networks, they are prone to failures, due to the way they work, and this affects the availability and management of files. One good way to handle files is by dividing them; when dealing with large size files, processes like uploading or downloading can take longer periods of time and also can consume more processing, because of this; it is more likely that a failure occurs. Provided that the hardware capabilities won't be enhanced during network operation, by employing a good partition technique could increase the chances of successful file treatment.[Wu et al., 2010].

Segmenting a file into pieces or segments is a process called chunking and each individual piece of a file is called chunk. Chunking can be performed using two different methods: the first is called file level chunking and as its name suggests, the whole file is handled as a chunk [Malhotra, 2015] and the second method, is called sub file chunking, in which a file can be divided in chunks of fixed or variable sizes[Min et al., 2011, Cai et al., 2013]. Each two of these methods has its own advantages and disadvantages. In fixed sized chunking, the file is segmented in chunks of the same size; even though it is easy and simple to execute, if a small piece of the file is modified or deleted, a new set of chunks must be generated; by contrast, variable size chunking segments a files in chunks of different sizes, utilizing boundaries which are based on the content of the file; although, its performance is higher compared to fixed size chunking, it requires longer time and more computational processing.

5.1. A stochastic method to divide a file in chunks

Dividing a file in chunks makes easier its storage and management within the network, in this part of the study a method to segment a file in chunks is proposed, based on the truncated geometric distribution.

5.1.1. Geometric distribution

A random variable X is defined by a Pascal or geometric distribution if the density of X is given by the equation:

$$f_X(x) = f_X(x,p) = \begin{cases} p(1-p)^x & x \in \mathbb{N} = \{1,2,...,n\} \ \ and \ p \in (0,1) \\ 0 & otherwise \end{cases}$$

A random variable X with geometric distribution is also called discrete waiting random variable, since it represents the number of failures that are expected before success, that why this distribution is used to model the number of Bernoulli trials[Ross, 2014].

In the figure **5-1**, it is displayed the behavior of the geometric distribution for different values for p, when this parameter is close to zero, the probability values calculated are very close with one another; nevertheless, when the parameter is close to 1, the probability values are very different with one another. This parameter p is of interest, provided that by adjusting its value it is possible to establish a level of heterogeneity or homogeneity between the chunks.

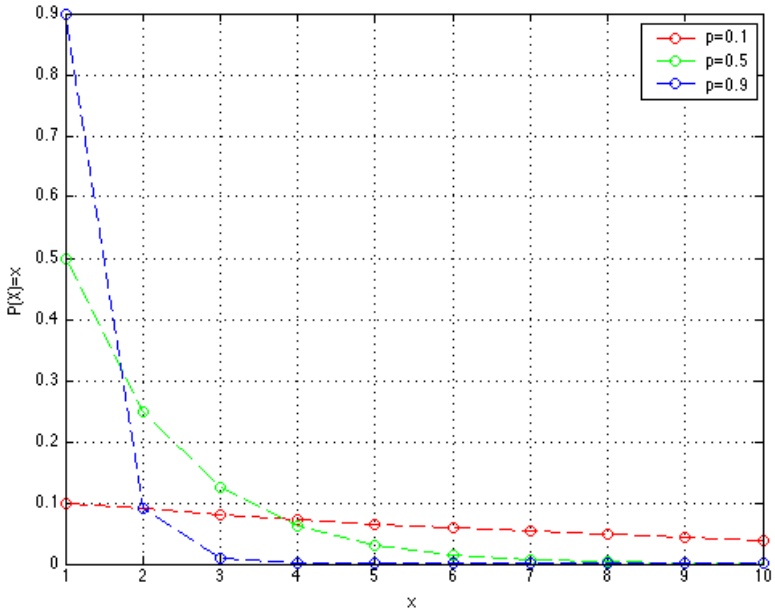


Figure 5-1.: Geometric distribution

5.1.2. Truncated geometric distribution

The domain of a random geometric variable is $\mathbb{N}$, however for some cases it is necessary to retrict this domain to a subset $A \subseteq \mathbb{N}$, which makes necessary to adjust the probability

structure of this random variable to the subset A. Now, consider Y a random variable with geometric distribution, with parameter p but truncated to a subset of A, as shown in equation 5-1:

$$f_Y(y) = \frac{f_X(y) I_{A(y)}}{P(A)} \tag{5-1}$$

Where I_A is the indicator function of subset A, and $P(A)$ can be calculated using equation 5-2:

$$P(A) = \sum_{x \in A} f_X(x) \tag{5-2}$$

Solving equation 5-1, it is obtained:

$$\begin{aligned} f_Y(y) &= \frac{f_X(y) I_{A(y)}}{P(A)} \\ &= \frac{p(1-p)^{y-1}}{\sum_{y=1}^{y=N} p(1-p)^{y-1}} \; where \; A = \{1, 2, ..., N\} \\ &= \frac{pq^{y-1}}{\sum_{y=1}^{y=N} pq^{y-1}} \; where \; q = 1 - p \\ &= \frac{pq^{y-1}}{p \sum_{y=1}^{y=N} q^{y-1}} \end{aligned}$$

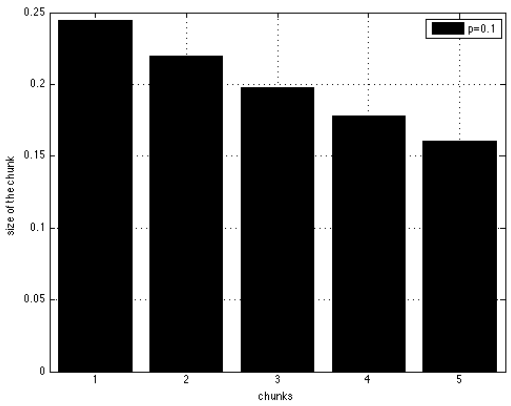
Using the geometric series over the denominator and simplifying, the probability density function for the truncated geometric distribution is found and is given by the equation 5-3.

$$f_Y(y) = \begin{cases} \frac{pq^{y-1}}{1-q^N} & y = \{1, 2, ..., N\} \\ 0 & otherwise \end{cases} \tag{5-3}$$

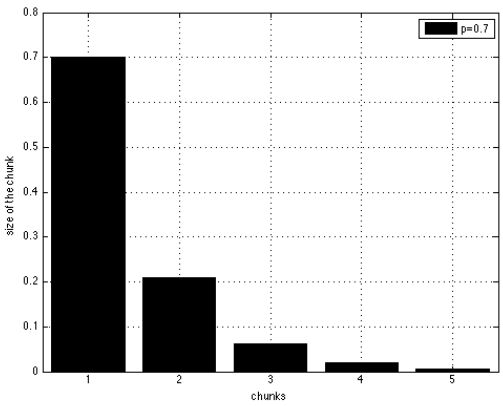
5.1.3. File segmentation using the truncated geometric distribution

The methodology to segment a file into chunks for this research will obey the truncated geometric distribution, because of the properties already mentioned previously. Consider that all of the chunks that conform a file X are assumed to be under the truncated geometric distribution, $X \sim G(P)$, then the size of each chunk which corresponds to a percentage of the whole file is calculated using the equation 5-3. To employ the truncated geometric distribution for file partition it is important to consider:

- Each file Y has a p value and a density function $F_Y(y)$ associated.



(a) File Y_1, $N_1 = 5$ and $p = 0.1$



(b) File Y_1, $N_1 = 5$ and $p = 0.7$

Figure 5-2.: Characterization of a chunk using the truncated geometric distribution.

- Each chunk that is part of a file Y is represented by the value y in $F_Y(y)$.
- Each chunk y in the file Y is part of the subset A, then $\|A\| = N$ and it represents the total number of chunks within the file.
- The probability value obtained after evaluating $F_Y(y)$ represents the percentage value of a chunk y, compared to the file Y, with this percentage the size of a chunk is calculated.
- To each file is associated a p value; this value represents the level of heterogeneity or homogeneity of the chunks sizes within the file Y. For one side, values for p closer to 1 will cause a high variation in the sizes of the chunks, on the other side p values closer to 0 will make their size similar.

In figures **5-2**a and **5-2**b, there is an example of the characterization of the a file Y_1 which has been divided into 5 chunks ($N = 5$). For one side, the level of heterogeneity is determined by the value of p, that in figure **5-2**a p is equal to 0.1, resulting in a similar size of the chunks, on the other side, in figure **5-2**b p is equal to 0.7 and the size of the chunks varies a lot.

After a file has been divided in chunks, a unique ID (fingerprint) must be added to each chunk[Malhotra, 2015]. This fingerprint is generated using hash algorithms such as MD5[Rivest, 1992] and SHA1[Commerce et al., 2012]. It is recommendable adding to each chunk fingerprint the total number of pieces that compose the file and the p parameter used to divide the file.

6. Simulation and model validation

Computers have become one of the main resources for research and are essential to analyze models through simulations, giving more options to verify the interactions between the components of the system and to analyze more amounts of data. Simulation is employed for theoretical and empirical research; it provides the means to explore all the capacities and boundaries of theoretical models and create synthetic conditions that would be hard to recreate on a real experiment. Simulation allows to make predictions about the expected behavior of a system which can be used as experimental set-up or as a support to make operation decisions. Simulation is also employed to study difficult and complex systems before spending resources on a real experiment[Thomas and Manz, 2017].

6.1. Output data analysis for a single system

When developing a model, a great amount of dedication and work is put on building and programming it; but, no so much to analyse its results, a very common practice is to run a simulation (replica) of an arbitrary length m and assume that its results describe the real characteristics of the system. Simulation models use random variables; therefore the output is random which makes a single replica useless. Since a simulation involves the realization of random variables that could have huge variances, then the result can differ greatly from the real system. Simulation can also be defined as a computer-based statistical experiment, and if its results will be used to validate a model, in order to give a good interpretation and meaning to the results it is very important to use appropriate statistical techniques[Law, 2007, Alexopoulos, 2007].

Let $x_{1,1}, \ldots, x_{1,i}, x_{1,m}$ be the realization from the output stochastic process $X_1, \ldots, X_i, X_m$ when using a set of random numbers as their input. If the same scenario is performed using a different set of random numbers as input, this will result in a different realization $x_{2,1}, x_{2,2}, x_{2,3}...x_{2,m}$ of the random numbers $X_1, X_2, \ldots, X_m$. Now, if n independent replications are performed in which the input parameters for the random numbers are reinitialized and the initial conditions are the same for each replication with a length m this will result in the observations shown in table **6-1**.

Table 6-1.: IID observations.

$x_{1,1},$	$x_{1,2},$	...	$x_{1,i},$	$x_{1,m}$
$x_{2,1},$	$x_{2,2},$	...	$x_{2,i},$	$x_{2,m}$
$\vdots$	$\vdots$		$\vdots$	$\vdots$
$x_{j,1}$	$x_{j,2}$	...	$x_{j,i}$	$x_{j,m}$
$\vdots$	$\vdots$		$\vdots$	$\vdots$
$x_{n,1},$	$x_{n,2}$	...	$x_{n,i},$	$x_{n,m}$

The observations from a single replica (row) cannot be processed with traditional statistical techniques, because they are auto correlated, no stationary, and not IID (Independent and identically distributed), consequently a replica of an arbitrary length has little significance by itself; nonetheless, note that the column i: $x_{1,i}, x_{2,i}, \ldots, x_{j,i}, x_{n,i}$ are IID observations for the random variable X_i. The basis for output data analysis for simulations is to perform n replicas each one of length m that share the same initial conditions; but using different seeds to produce random numbers and finally use the IID observations $x_{j,i}$ (where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$) to gain information to estimate performance measures for he behavior of the system.

6.1.1. Transient and steady state behavior of a stochastic process

Consider $X = X_1, X_2, \ldots, X_m$ the output of a stochastic process, now let $F_i(X \mid I) = P(X_i \leq x \mid I)$, where $F_i(X \mid I)$ at time i given the initial conditions I.

As can be seen in figure **6-1** each transient distribution has a density function f_{y_i}, the density functions specify how the behavior of the random variable changes from one replication to another. If x and I are fixed then, $F_1(x \mid I)$, $F_2(x \mid I)$, ...$F_i(x \mid I)$ will be just a sequence of numbers; if $F_i(X \mid I) \to F(x)$ as $i \to \infty$ for every x and I, then $F(x)$ is called the steady-state distribution of the output process X. As can be understood the steady-state distribution $F(x)$ occurs at a point in which $i \to \infty$ or i is sufficiently large, as shown in figure **6-1**, there is time $k+1$ where steady state starts. Please keep in mind that steady state does not imply that the random variables after X_{k+1} will have the same value; instead it means they will have approximately the same distribution. Additionally these random variables won't be independent; rather they will form a covariance-stationary stochastic process.

6.1.2. Types of simulation according to the output analysis

According to the way of finishing a simulation, two types can be found: terminating and non-terminating simulations. In Terminating simulations (also called transients) the short-run behavior of a system is studied, the performance measure of interest is estimated within a

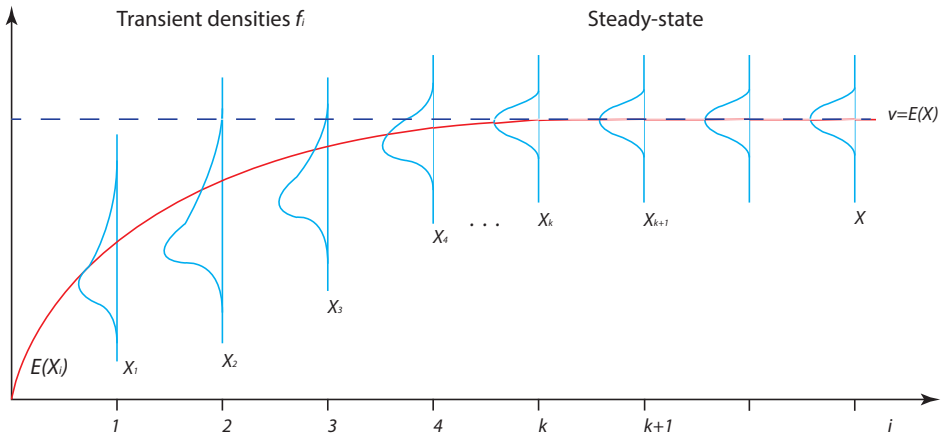


Figure 6-1.: Transient and steady-state density functions for a stochastic process.

period whose end is marked by an event E, which can be deterministic, for instance ($E = 20$) seconds or random, for example when the number of jobs in a queue reaches 500 ($E = 500$). Usually the nature of the problem defines E.

Non terminating simulations (steady-state) aim to study the long-run behavior of a system, which starts at $i = 0$ and converges when $i \to \infty$ or is large enough; this means that there is not any event E that specifies when a simulation finishes; however, in a practical simulation, the researcher defines its duration in such way that it allows to obtain good estimates of interest. These types of simulations are employed when designing new systems or making changes over an existing one.

Both types of simulation depend a lot on the initial conditions, since they have some impact on the results and may led to errors, therefore care must be taken when selecting initial conditions taking into consideration that they must be representative of those of the actual real system.

Statistical analysis for steady state parameters

Consider ϕ as a steady-state parameter that is characteristic of X like $E(X)$, the estimation of ϕ causes a problem when the distribution of X_i is different from F, due to the initial conditions, the initial output data is not very representative of such behavior, raising a question about how to choose simulation output data that actually represents the steady

state behavior. Due to this, the estimators of ϕ from some initial observations may not be representative. This situation is called the problem of the initial transient or the start-up problem. One technique commonly employed to face this situation is named warming-up the model or initial data deletion, which goal is to identify an index l such that $(1 \leq l \leq m-1)$, deleting the observations $X_1, X_2, \ldots, X_l$ and finally using the remaining observations to estimate v shown in equation 6-1.

$$\bar{X}(m,l) = \frac{1}{m-l} \sum_{i=l+1}^{m} X_i \tag{6-1}$$

Since $X(m,l)$ does not consider the observations until l which may have been affected by the initial conditions, it is likely to be less biased than $X(m)$; nevertheless, m and l must be chosen in such way that $E\left[\bar{X}(m,l)\right] \approx v$. If they are chosen too small $E\left[\bar{X}(m,l)\right]$ may significantly different than v on the opposite if they are chosen too large, $E\left[\bar{X}(m,l)\right]$ will have an excessive variance.

One technique broadly used to find the index l such that $E[X_i] \approx v$ for $i > l$ is the Welch graphical method. Provided that a single replication is not enough to determine l, this method uses multiple n replications and works as follows:

1. Make n replication, each one of length m, where m is large.
2. Across the replicas compute $\bar{X}_i = \sum_{j=1}^{n} \frac{X_{j,i}}{n}$ for $i = 1, 2, \ldots, m$.
3. In order to soften the high-frequency oscillations from the previous step, the method uses a moving average $\overline{Y}_i(w)$ where w is the time window and is defined as follows:

$$\bar{X}_i(w) = \begin{cases} \frac{1}{2w+1} \sum_{s=-w}^{w} (\bar{X}_{i+s}) & \text{for} \quad i = w+1, \ldots, m-w \\ \frac{1}{2i-1} \sum_{s=1-i}^{i-1} (\bar{X}_{i+s}) & \text{for} \quad i = 1, \ldots, w \end{cases} \tag{6-2}$$

   it is recommended that $w \leq \frac{m}{4}$.

4. Plot the $\overline{X}_i(w)$'s, if the curve is reasonably smooth choose a value for l at a point after which $\overline{X}_1(w), \overline{X}_2(w), \ldots$ seems to have converged, otherwise pick another value for w and repeat the whole procedure again, then if the response is not satisfactory add more replicas and carry out the whole procedure again.

The replication-deletion approach

The replication-deletion approach is a method proposed in[Law, 2007] to obtain a point estimate and confidence interval for v which offers the next advantages:

Table 6-2.: Observations from a n replication simulation of length m

$X_{1,1}$	$X_{1,2}$	...	$X_{1,l}$	$X_{1,l+1}$	...	$X_{1,m}$
$X_{2,1}$	$X_{2,2}$	...	$X_{2,l}$	$X_{2,l+1}$	...	$X_{2,m}$
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$
$X_{n,1}$	$X_{n,2}$	...	$X_{n,l}$	$X_{n,l+1}$	...	$X_{n,m}$

- When used correctly, it has a good statistical performance.
- It is easy to understand and implement.
- It can be applied to all types of output parameters and to make different estimates.
- It is useful to make comparison between different system configurations.

Suppose that n independent replications, each one of length m were performed and that l has been already estimated using the Welsch graphical method, resulting in the observations in **6-2**.

The first "l" ($m \gg l$) observations in each replication can be deleted since they are not representative of the steady-state behavior; with the remaining $X_{j,l+1}, \ldots, X_{j,m}$ let Y_j be defined by:

$$Y_j = \frac{1}{m' - l} \sum_{i=l+1}^{m'} X_{j,i} \quad \text{for} \quad j = 1, 2, \ldots, n' \tag{6-3}$$

Note that the $Y_j s$ are IID observations which can be used with classical statistics to build a point estimate and confidence interval for v. Let the sample mean be given by:

$$\overline{Y}(n') = \frac{1}{n'} \sum_{j=1}^{n} Y_j \tag{6-4}$$

and the sample variance:

$$S^2(n') = \frac{1}{n' - 1} \sum_{j=1}^{n} (Y_j - \overline{Y}(n'))^2 \tag{6-5}$$

Thus for v an approximate $100(1 - \alpha)$ percent confidence interval is given by:

$$\overline{Y}(n') \pm t_{n'-1,1-\frac{\alpha}{2}} \sqrt{\frac{S^2(n')}{n'}} \tag{6-6}$$

6.1.3. Comparing two systems

A very common practice in simulation is to use its results to compare the outcome of two or more systems with different configurations based on some performance measures, since simulation works as support to make decisions, by comparing two systems, more support is gained before the real experimentation or operational implementation.

Even though the statistical techniques explained during this chapter make use of IID observations to estimate different parameters, when comparing systems usually the same seeds are assigned to the design points of interest, resulting in a fair comparison if the systems are under the same experimental conditions, this technique is known as common random numbers[Banks, 1998] (CRN). For the case of simulation, is the same source of random numbers, which is achieved thanks to the simulation software employed, by synchronizing the random numbers.

A paired-t confidence interval to compare steady-state measures of performance

Consider a comparison problem between two systems $k = 1, 2$ regarding their performance. In order to give a solution to this problem confidence intervals are built. Assume that both simulation are driven using CRN and the same number of replicas $n_k = n$. Let μ_k be the steady-state measure of interest, then it is desired to build a confidence interval for $\zeta = \mu_1 - \mu_2$.

Define $Z_i = Y_{1,i} - Y_{2,i}$ for $i = 1, 2, \ldots, n$ and $\zeta = E(Z_i)$ then:

$$\overline{Z}(n') = \frac{1}{n'} \sum_{i=1}^{n'} Z_i \tag{6-7}$$

And the variance:

$$\widehat{VAR}\left[\overline{Z}(n')\right] = \frac{1}{n'(n'-1)} \sum_{i=1}^{n'} \left[Z_i - \overline{Z}(n')\right]^2 \tag{6-8}$$

A $100(1-\alpha)$ percent confidence interval for ζ is given by:

$$\overline{Z}(n') \pm t_{n'-1,1-\frac{\alpha}{2}} \sqrt{\widehat{VAR}\left[\overline{Z}(n')\right]} \tag{6-9}$$

The confidence interval is valid and covers ζ with a probability of $(1-\alpha)$ percent if the $Z_i s$ are normally distributed, note that it was not assumed independence between $Y_{1,i}, Y_{2,i}$, neither their variances were the same; this allows positive correlation between $Y_{1,i}$ and $Y_{2,i}$ along with the use of CRN leading to a reduction in the $\widehat{VAR}\left[\overline{Z}(n')\right]$ and thus a shorter confidence interval; as it becomes smaller, it facilitates to detect differences between the systems.

Table 6-3.: 2^2 factorial design matrix

Factor combination	Factor 1	Factor 2	Response
1	$'-'$	$'-'$	R_1
2	$'+'$	$'-'$	R_2
3	$'-'$	$'+'$	R_3
4	$'+'$	$'+'$	R_4

6.2. Sensitivity analysis

While analysing the behavior of a system, it is of interest to find out what happens when changes are made to the input parameters and how this impacts a performance measure. To estimate the change for a simulation outcome as the input parameters vary, sensibility analyses are employed.

There are two types of sensibility analysis techniques: local and global. Local techniques are performed one factor at the time, changing one while keeping the others fixed, while global techniques explore the definition interval of each factor, in which the impact of each factor is an average over the possible values of the other factors[Saltelli, 2004]. For this study it is of interest global techniques, in particular screening strategies, in which the influence exercised by the factors in the response is evaluated, filtering non-useful factors as possible. Provided that in this study few factors are being considered, following the recommendation in[Law, 2007], in this study factorial design techniques will be used. Factorial designs techniques are used in experimental design[Kleijnen, 2005] in order to gain an insight into the system's behavior with a reasonable quantity of factor combinations. Within the factorial design techniques, there can be found 2^k factorial designs.

6.2.1. 2^k factorial design

Presume that a model has $k \geq 2$ factors and it is desired to estimate the impact of each factor on the response and also if the factors interact with each other; in order to achieve this 2^k factorial design is used, in this technique, two levels for each k factor are selected and them each 2^k possible combination is simulated. To identify the levels a $-$ and $'+'$ symbols are used; nonetheless, specifying them requires the knowledge of the analyst in order to assign them reasonable values; as suggested by the signs the levels should be opposite of each other but not to the point of being at unrealistic extremes. The experiment can be represented using a table, for example for $k = 2$ it would be as shown in table **6-3** also referred as design matrix.

Each response is the result of a simulation when a combination of factors is at its respective levels $-$ or $+$. The impact of a factor k is the average change in the response due to the

change from $-$ to $+$ while keeping the other factors fixed. This average considers every combination of the oher $k-1$ factors. Note that the main effect is determined with respect to the current design and factors, therefore it is not possible to make extrapolations if other conditions are not fulfilled. To calculate the main effect of a factor, apply the signs in the factor k column to the response, add them up and divide by 2^{k-1}, for example using the information from design matrix **6-3**, the effect for factor e_1 will be defined by:

$$e_1 = \frac{-R_1 + R_2 - R_3 + R_4}{2} \tag{6-10}$$

And rewritten:

$$e_1 = \frac{(R_2 - R_1) + (R_4 - R_3)}{2} \tag{6-11}$$

In some cases the level of a factor k_1 may depend on the level of another factor, say k_2, in this case, these factors interact and the interaction effect is defined by half the difference between the average effect of factor k_1 when factor k_2 is at $+$ minus the average effect of factor k_1 when factor k_2 is at $-$, for example for $e_{1,2}$ will be defined by:

$$e_{1,2} = \frac{R_4 - R_3}{2} - \frac{R_2 - R_1}{2} \tag{6-12}$$

It can be calculated by multiplying the signs of both factors and then repeating the same procedure explained for the main effect. Note that in the design matrix **6-3**, in the second half that factor k_2 is $+$ while factor k_1 is moving from $-$, to $+$ therefore the first half of the expression 6-12 reflects the average of moving factor k_1 from $-$, to $+$ when factor k_2 remains constant at $+$, similarly the second half of the expression 6-12 shows the effect of moving factor k_1 from $-$, to $+$ while factor k_2 remains at $-$. Then the difference between these two parts of the expression is the difference effect that factor k_1 exercises on the response depending on the levels of factor k_2. As can be deduced the effect is symmetric, so $e_{1,2} = e_{2,1}$.

A three-factor interaction is possible and is obtained in similar fashion as the two factor interaction; nevertheless its interpretation is more difficult. If there are higher interactions, the effects cannot be interpreted as the change from $-$, to $+$ since the magnitude and change depend at least on the level of another factor, under this situation the experiment needs to be interpreted in a different manner.

As explained during this chapter, a single replica is not enough, in order to determine if an effect is real, it is necessary to estimate its variance, a very common practice in simulation experiments is to execute n replicas of each combination of the design matrix to obtain n independent values for each effect, then using the t distribution along with these results, a $100(1-\alpha)$ confidence interval is built for each effect with $n-1$ df. If the confidence interval

for a given effect does not include 0, then this effect is real, otherwise the statistical evidence suggest that it is not present[Law, 2014].

6.3. Simulation procedure

Simulation along with a proper analysis of the output data will be the methodologies employed to validate the behavior of the theoretical model developed during this study, the simulation will be carried out on the NS-3 network simulation and the output data analysis will follow the methodologies from this chapter.

6.3.1. NS-3 Network simulator

NS-3 is a discrete event driven network simulator[Carneiro, 2010, Kamoltham et al., 2012], an open source software licensed under GNU GPLV2, it is written in C++ with bindings for python with simulations in C++ executable or python programs and principally supported by Linux, OSX and Free BSD, NS-3 offers an environment to design, test and improve network models and protocols. In the recent years NS-3 has become one of the prominent network simulator, targeted mainly towards academic and research use.

6.3.2. Testing environment and assumptions

All the simulation experiments were performed considering the general parameters found on table **6-4** conditions and the next assumptions:

- All the nodes move freely.
- The nodes have a limited amount of disk space to store data files.
- On ns-3 files cannot be uploaded, neither managed, therefore, in the simulation they will be treated as traffic; to simulate that the nodes have a hard drive, a counter variable will decrement or increment according to the traffic received or sent.
- The files are always consistent.
- During each simulation half of the nodes are chosen randomly to store original chunks of a file.
- Each node has a battery of limited capacity.
- The cost for energy and disk space is equal to 1 and the agents are price-takers.

Table 6-4.: General simulation parameters

Parameter	Characteristics
Geographic space	Flatland
Number of nodes	36
Propagation model	NS-3 Constant
Loss model	NS-3 Two Ray Ground Propagation Loss Model
Mobility model	Random Direction 2d MobilityModel
Simulation time	600 seconds
Energy source	Basic Energy Source
Energy model	Simple Device Energy Model
Version	3.24.1

Table 6-5.: Testing scenario parameters.

Model Parameter	Scenario 1	Scenario 2
Energy (units)	100	20-100
Disk Space (units)	100	20-100
File size	10240	10240
p parameter	0.1	0.1
Molecules	100-200	100-200
Number of hops	1	1
Molecules capacity	10000	10000
QS threshold	0.51	0.51
Mutation probability	0.1	0.1
Clonning probability	0.1	0.1

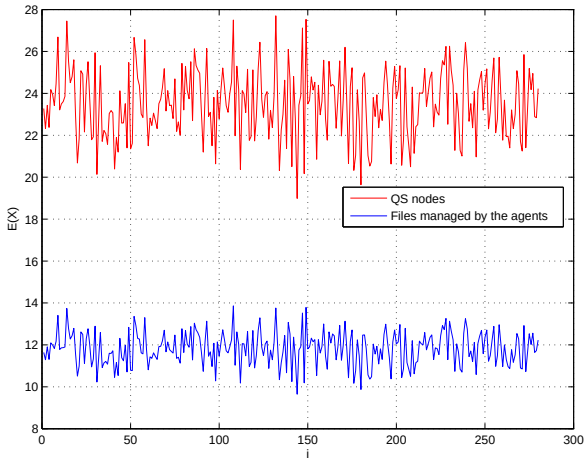
In order to evaluate the performance of the model proposed during this dissertation, two different testing scenarios were proposed. In the first scenario all the nodes have the same hardware capabilities and they are at their maximun values, on the contratry in the second scenario all the nodes have different hardware capabilities, the description of each scenario is showed in detatil in table **6-5**. Note that the scenarios use CRN.
After simulation, two results will be considered:

1. the quantity of nodes induced to QS state by the b-agents.

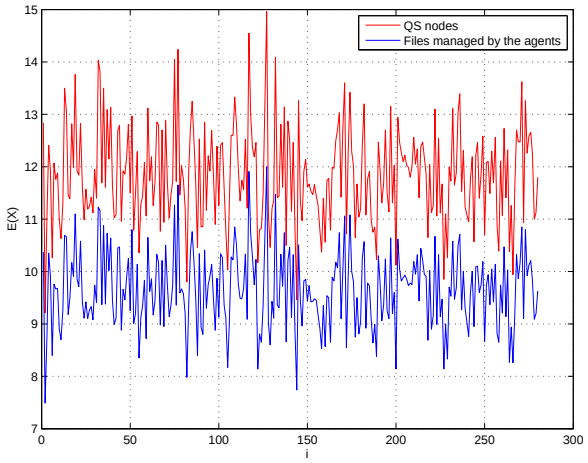
2. the quantity of files managed by agents.

6.3.3. Output data analysis and comparison between systems

The next procedures were done in order to obtain the data to analyse:



(a) Average response for scenario 1.



(b) Average response for scenario 2.

Figure 6-2.: Response of the simulation scenarios.

1. Following the indications supplied in [Law, 2007], for each scenario were carried out $n = 100$ independent replications of simulation experiments each one of length $m = 280$.

2. Use the Welsch graphical method to determine the moment at which the steady-state behavior begins.

3. Use the Replication-Deletion approach[Law, 2007] to estimate the steady-state mean given a confidence interval of 90

4. Use a $paired - t$ confidence interval to make comparisons between the two system configurations.

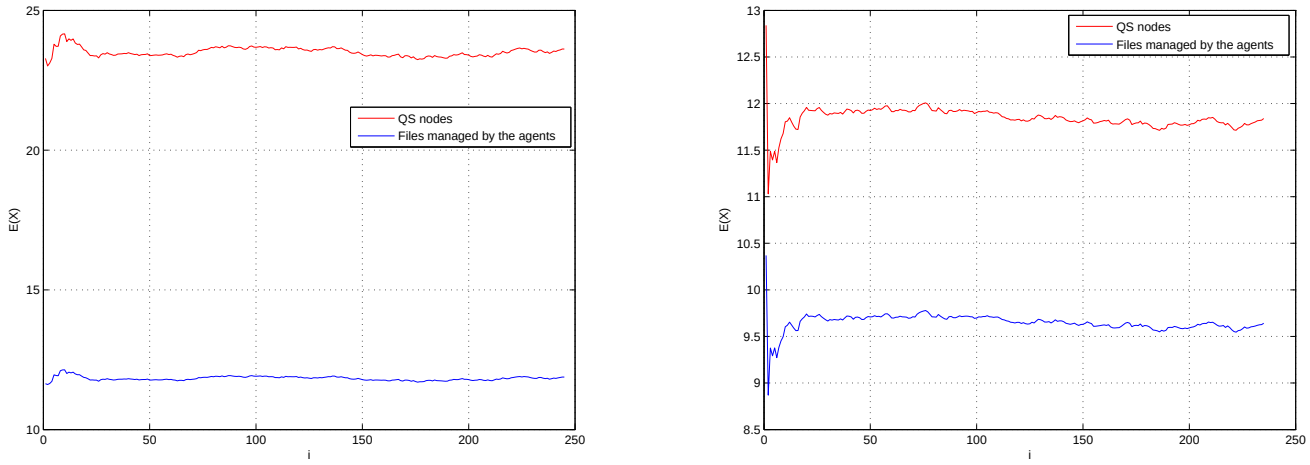
Results

1. After making the initial independent simulation the average response for each scenario was plotted in figures **6-2**a and **6-2**b.

2. Using the Welch graphical method with a window value $w_1 = 40$ and $w_2 = 45$ for each scenario respectively, obtaining the results showed in plots **6-3**a and **6-3**b.

3. By graphical inspection, it is possible to see that both plots start to converge at $l_{sc1} = 25$ and $l_{sc2} = 32$ for each scenario respectively.

4. After applying the replication deletion approached, There is a 90 % of confidence that the mean for the nodes induced to QS and the files managed by the agents are between the values shown in table **6-6**.



(a) Moving average for scenario 1.

(b) Moving average for scenario 2.

Figure 6-3.: Moving averages.

Table 6-6.: Steady-state parameters for both scenarios.

Scenario	Responce	Point Estimate	Variance	Confidence interval
Scenario 1	Quorum sensing nodes	23.468 ± 0.131	1.035	$[23.337, 23.599]$
	Files managed by the agents	11.809 ± 0.065	0.252	$[11.745, 11.874]$
Scenario 2	Quorum sensing nodes	11.856 ± 0.0765	0.352	$[11.780, 11.933]$
	Files managed by the agents	9.657 ± 0.060	0.220	$[9.596, 9.717]$

Table 6-7.: 90 % confidence interval for the difference of the mean.

	Point estimate	Variance	Confidence interval
QS nodes	11.612 ± 0.141	1.194	$[11.471, 11.753]$
Files	2.153 ± 0.080	0.388	$[2.072, 2.233]$

Table 6-8.: Factor levels.

Factor	Low-level	High-level
Molecules capacity	5000	15000
Quorum threshold	0.3	0.7
Cloning probability	0.05	0.15
Mutation probability	0.05	0.15

5. With a 90 % of confidence there is enough evidence to support that the means for the nodes induced to QS and the files managed by the agents is higher in the first scenario than the second. The difference of means are likely to be between the values shown in table **6-7**.

6.3.4. Sensitivity analysis

In order to estimate how the imput parameters of the model impact the response of each system configuration, sensibily analysis were used under the factor levels of table **6-8**.

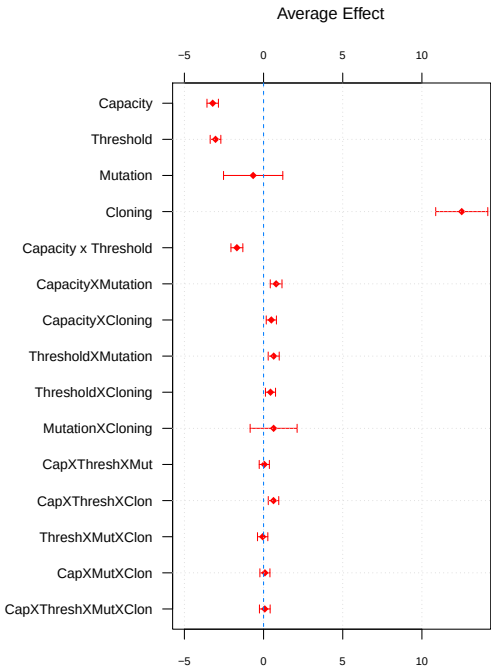
After simulating using 2^4 factorial design and making 100 replicas for each combination, obtaining the results showed in table **6-9**. The main effects and the interactions for the first scenario are shown in figures **6-4**a and **6-4**b for the nodes and the files respectively. And for the second scenario shown in figures **6-5**a and **6-5**b. By looking at the plots and the design matrix for the effects, it is pretty clear that the factor that has the greatest impact on the response of the system is the cloning probability; the reason for this is in the QS; provided it relies con population density, when the cloning probability is low, there are few agents, probably the initial population plus some clones, therefore there are low amounts of molecules in each node; nevertheless, when the cloning probability increases, so does the number of agents and the amount of molecules in each node. This shows that in order to increase the success of the communication strategy proposed in this research, the population of agents needs to grow in a scalable manner.
Additionally two factors stand out, the molecules capacity and the quorum threshold, which have similar effects in both scenarios. When both of these factors increase their impact on the response is negative, since both require that more molecules be released in each node. By detailing it can be seen that the impact is greater on scenario two than in scenario one. Scenario two is very heterogeneous and each due to the decision making mechanism employed

each node may find itself in a different utility curve and because of the budget restrictions (molecules) it is hard for an agent to find nodes that are on the same utility curve that maximizes the agent's income (molecules), while on the scenario 1 all the nodes are on the same utility curve it is easier for an agent to find nodes fulfill its utility, hence a single agent while traversing the network can release molecules on several nodes.

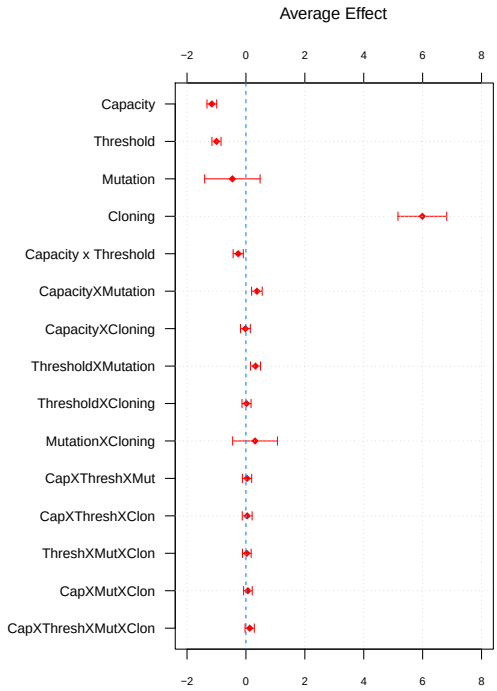
The last factor is the mutation probability, but as can be seen in the plots, this is the factor that has the lowest effect on the response, provided this result this factor must be redesigned for future implementations of the communication strategy proposed in this investigation. Note that in the majority of the plots the interaction does not have a significant effect on the response, however for the amount of nodes in the scenario one in plot **6-4**a and for the amount of files in scenario 2 in plot **6-5**b there is two interaction between the amount of molecules and the quorum threshold, this means that the effect either factor depends on the level of the other. To check the information of the effects and interactions in detail, please refer to table **6-10**.

					Scenario 1				Scenario 2			
Run	Molecules Capacity	Quorum threshold	Mutation Probabilty	Cloning Probability	Nodes	Variance	Files	Variance	Nodes	Variance	Files	Variance
1	-1	-1	-1	-1	22.28	290.547	11.28	69.173	14.83	141.112	10.68	70.058
2	1	-1	-1	-1	20.15	310.795	10.05	77.199	11.05	89.098	9.1	57.990
3	-1	1	-1	-1	20.39	316.362	10.18	78.715	11.48	92.535	9.54	62.958
4	1	1	-1	-1	13.7	192.232	8.52	64.535	5.73	40.381	4.97	30.191
5	-1	-1	1	-1	19.55	310.371	9.82	73.866	15.68	144.321	10.97	68.029
6	1	-1	1	-1	18.9	320.677	9.38	78.985	10.95	86.876	8.92	55.973
7	-1	1	1	-1	19.12	319.359	9.49	77.909	11.86	100.162	9.45	61.826
8	1	1	1	-1	13.78	204.658	8.26	67.528	6.04	43.029	5.15	30.048
9	-1	-1	-1	1	33.81	58.782	16.97	12.615	21.4	66.081	15.32	28.684
10	1	-1	-1	1	31.39	126.927	15.74	28.780	16.38	49.228	13.41	30.628
11	-1	1	-1	1	31.81	111.731	16.02	25.212	16.93	52.025	14.21	33.218
12	1	1	-1	1	27.03	199.060	14.02	49.495	13.18	41.806	10.9	27.828
13	-1	-1	1	1	32.45	103.604	16.19	24.155	22.52	44.777	15.97	18.130
14	1	-1	1	1	31.54	137.301	15.51	33.424	17.35	40.513	14.09	24.366
15	-1	1	1	1	31.37	139.064	15.62	34.016	18.23	37.573	14.8	21.253
16	1	1	1	1	28.59	161.355	14.83	38.304	14.13	41.690	11.28	24.345

Table 6-9.: 2^4 factorial matrix design.

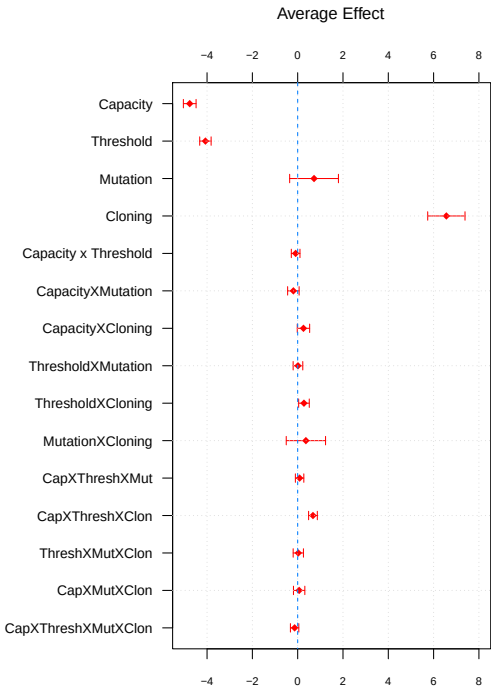


(a) Effect on average QS nodes.

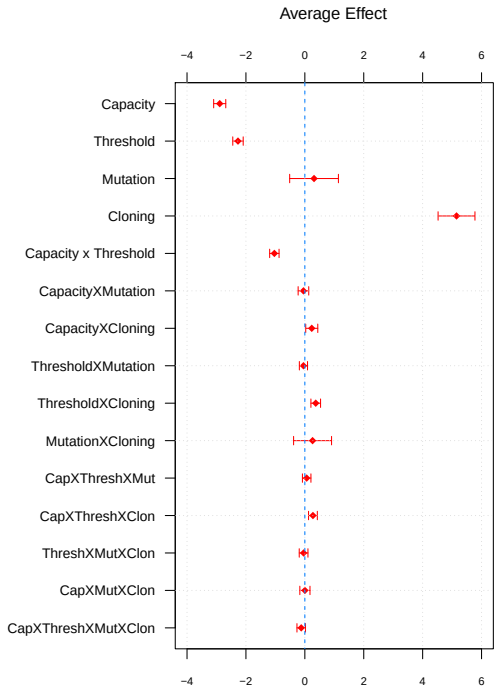


(b) Effect on average files.

Figure 6-4.: Effects for the scenario 1.



(a) Effect on average QS nodes.



(b) Effect on average files.

Figure 6-5.: Effects for the scenario 2.

Table 6-10.: Main effects and interactions with a 90 % of confidence.

	Scenario 1				Scenario 2			
Factor	Nodes	Variance	Files	Variance	Nodes	Variance	Files	Variance
Molecules C	-3.213±0.361	7.842	-1.158±0.166	1.657	-4.765±0.280	4.718	-2.890±0.205	2.534
Quorum T	-3.035±0.337	6.807	-1.000±0.154	1.431	-4.073±0.251	3.792	-2.270±0.177	1.880
Mutation P	-0.658±1.871	210.451	-0.460±0.941	53.194	0.723±1.076	69.634	0.313±0.827	41.135
Cloning P	12.515±1.645	162.565	5.990±0.828	41.183	6.563±0.823	40.720	5.150±0.626	23.546
Molecules C X Quorum T	-1.685±0.375	8.465	-0.263±0.172	1.783	-0.090±0.191	2.189	-1.035±0.159	1.528
Molecules C X Mutation P	0.793±0.370	8.244	0.373±0.179	1.936	-0.190±0.253	3.860	-0.048±0.180	1.943
Molecules C X Cloning P	0.490±0.322	6.216	-0.018±0.166	1.653	0.255±0.274	4.497	0.235±0.205	2.517
Quorum T X Mutation P	0.640±0.348	7.280	0.325±0.170	1.746	0.013±0.213	2.718	-0.048±0.139	1.153
Quorum T X Cloning P	0.438±0.318	6.066	0.020±0.153	1.412	0.278±0.234	3.293	0.370±0.164	1.609
Mutation P X Cloning P	0.635±1.483	132.138	0.310±0.760	34.718	0.363±0.867	45.217	0.263±0.645	24.966
Molecules C X Quorum T X Mutation P	0.045±0.320	6.161	0.038±0.157	1.490	0.085±0.186	2.086	0.063±0.144	1.247
Molecules C X Quorum T X Cloning P	0.628±0.331	6.594	0.043±0.168	1.686	0.675±0.192	2.216	0.275±0.149	1.338
Quorum T X Mutation P X Cloning P	-0.058±0.326	6.399	0.030±0.147	1.294	0.028±0.227	3.087	-0.043±0.148	1.318
Molecules C X Mutation P X Cloning P	0.085±0.338	6.859	0.068±0.156	1.468	0.065±0.185	2.059	0.003±0.145	1.261
Molecules C X Quorum T X Mutation P X Cloning P	0.078±0.317	6.046	0.128±0.146	1.282	-0.135±0.251	3.771	-0.123±0.172	1.780

7. Conclusions and recommendations

7.1. Conclusions

1. Due to the massive utilization of wireless mobile devices and the current growth in mobile computation, it is necessary to develop techniques that ensure data availability provided the special circumstances of MANET functioning. Ensuring data availability is a huge challenge and the main property for contents management in MANETs, which is the main reason to develop the proposed method.

2. Considering that during operation it is not feasible to upgrade the MANET resources and due to their constant utilization they become sparse, this makes possible applying concepts from microeconomics, which allow each agent to go rationally after a goal that maximizes its utility, make decisions based on its own interests and the node conditions; by making individual decisions a global behavior emerges which also helps tuning the response of the QS.

3. A methodology to segment files based on the truncated geometric distribution was proposed, which allows characterizing the chunks that belong to a file. This method makes possible to define a level of heterogeneity of the chunks by adjusting the p value of the distribution.

4. Quorum sensing is a good mechanism for large-scale synchronization to achieve collective behavior; by applying it as a mean of communication to a MAS interacting over a MANET, as expected it helped agents to communicate and coordinate with each other. Once QS has been attained, task associated with the services of the network are much easier to execute.

5. Quorum Sensing adds additional information to the environment, it is a method to spread local information which allowed to gain a local knowledge of the current state of local neighbourhoods without having direct communication between the nodes; however, in order to be effective for the agents, the information must be enough, since the agents will never have the complete information of the network, they depend completely on the information they can extract from the environment, with the QS information they can map the system and help to build and maintain the information of their local environment.

6. Cloning is a very important factor for the strategy of this study as was seen in the results this factor exercises a great effect over the average response of the model additionally it brings enough evidence to support that the quorum sensing relies on the population density in order to be achieved.

7. Simulation is a good methodology to validate theoretical models; nonetheless, it is important that in order to use the results of a simulation to obtain performance measure of the system it is necessary to carry out appropriate statistical analysis, as evidenced in this research those procedures are easy to understand as long as they have appropriate data to process.

7.2. Future work

- Propose methodologies to define the prices of the goods of the network and the income of the agents, considering the producers theory and including negotiation strategies between the parts, in a similar way as a market of goods and services would behave in real life.

- Characterize a mechanisms to carry out cloning in a manner similar to the bacteria reproduction, additionally perform cell specification to create new agents that can have perform different tasks according to their role and also propose a strategy to perform communication between agents with different roles .

A. Appendix: NS-3 code

In the cd in which this dissertation was attached, there is a $C++$ file named GeneralCode-ForSimulation in which as its name suggests, there is a code to reproduce all the simulations carried out during this dissertation.

Bibliography

[Akyildiz et al., 2002] Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422.

[Akyildiz et al., 2005] Akyildiz, I. F., Wang, X., and Wang, W. (2005). Wireless mesh networks: A survey.

[Alexopoulos, 2007] Alexopoulos, C. (2007). Statistical analysis of simulation output: state of the art. *Proceedings of the 2007 Winter Simulation Conference*, 100(Ci):150–161.

[Amaris et al., 2015] Amaris, J. E. P., Hurtado, A. C. C., and Trivino, J. E. O. (2015). Bacteria agent colony inside an ad-hoc network. In *2015 10th Computing Colombian Conference (10CCC)*, pages 347–350. IEEE.

[Babaoglu et al., 2006] Babaoglu, O., Canright, G., Deutsch, A., Di Caro, G. a., Ducatelle, F., Gambardella, L. M., Ganguly, N., Jelasity, M., Montemanni, R., Montresor, A., and Urnes, T. (2006). Design Patterns from Biology for Distributed Computing. *ACM Transactions on Autonomous and Adaptive Systems*, 1(1):26–66.

[Banks, 1998] Banks, J. (1998). Handbook of simulation: Principles, methodology, advances, applications, and practice. *Methodology Advances*, 32:849.

[Bassler, 2002] Bassler, B. L. (2002). Small talk: Cell-to-cell communication in bacteria. *Cell*, 109(4):421–424.

[Besanko and Braeutigam,] Besanko, D. and Braeutigam, R. R. *Microeconomics*.

[Cai et al., 2013] Cai, B., Zhang, F. L., and Wang, C. (2013). Research on chunking algorithms of data de-duplication. *Advances in Intelligent Systems and Computing*, 181 AISC:1019–1025.

[Carneiro, 2010] Carneiro, G. (2010). Ns-3: Network simulator 3. In *UTM Lab Meeting April*, volume 20.

[Chlamtac et al., 2003] Chlamtac, I., Conti, M., and Liu, J. J.-N. (2003). Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, 1(1):13–64.

[Commerce et al., 2012] Commerce, U. S., , and Technology, N. I. (2012). Secure Hash Standard (SHS). *Federal Information Processing Standards Publication 180-4*, (October):36.

[Conti et al., 2004] Conti, M., Gregori, E., and Maselli, G. (2004). Cooperation issues in mobile ad hoc networks. *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on*, pages 803–808.

[Deb et al., 2002] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Trans. Evol. Comp*, 6(2):182–197.

[Eberhard et al., 1981] Eberhard, A., Burlingame, a. L., Eberhard, C., Kenyon, G. L., Nealson, K. H., and Oppenheimer, N. J. (1981). Structural identification of autoinducer of Photobacterium fischeri luciferase. *Biochem.*, 20(9):2444–2449.

[Ehrgott and Gandibleux, 2002] Ehrgott, M. and Gandibleux, X. (2002). Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys. *Multiple Criteria Optimization*, page 496.

[Federle and Bassler, 2003] Federle, M. J. and Bassler, B. L. (2003). Interspecies communication in bacteria. *The Journal of clinical investigation*, 112(9):1291–9.

[Fitzek and Katz, 2006] Fitzek, F. H. P. and Katz, M. D. (2006). *Cooperation in wireless networks: Principles and applications: Real egoistic behavior is to cooperate!*

[Friedman et al., 2010] Friedman, R., Kliot, G., and Avin, C. (2010). Probabilistic quorum systems in wireless Ad Hoc networks. *ACM Transactions on Computer Systems*, 28(3):1–50.

[Gilbert et al., 2010] Gilbert, S., Lynch, N. a., and Shvartsman, A. a. (2010). Rambo: A robust, reconfigurable atomic memory service for dynamic networks. *Distributed Computing*, 23:225–272.

[Gobbetti et al., 2007] Gobbetti, M., De Angelis, M., Di Cagno, R., Minervini, F., and Limitone, A. (2007). Cell-cell communication in food related bacteria. *International Journal of Food Microbiology*, 120(1-2):34–45.

[Hara, 2001] Hara, T. (2001). Effective replica allocation in ad hoc networks for improving data\naccessibility. *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, 3:1568–1576.

[Hayashi et al., 2005] Hayashi, H., Hara, T., and Nishio, S. (2005). A Replica Allocation Method Adapting to Topology Changes in Ad Hoc Networks. pages 868–878.

[Hoebeke J., 2004] Hoebeke J., M. I. D. B. D. P. (2004). An overview of mobile ad hoc networks: Applications and challenges. *Journal of the Communications Network*, 3(3):60–66.

[Hoolimath et al., 2012] Hoolimath, P. K. G., Kiran, M., and Reddy, G. R. M. (2012). Optimized Termite: A bio-inspired routing algorithm for MANET's. In *2012 International Conference on Signal Processing and Communications (SPCOM)*, pages 1–5. IEEE.

[Jayaraman and Wood, 2008] Jayaraman, A. and Wood, T. K. (2008). Bacterial quorum sensing: signals, circuits, and implications for biofilms and disease. *Annual review of biomedical engineering*, 10:145–167.

[Kamoltham et al., 2012] Kamoltham, N., Nakorn, K. N., and Rojviboonchai, K. (2012). From NS-2 to NS-3 - Implementation and evaluation. *2012 Computing, Communications and Applications Conference, ComComAp 2012*, pages 35–40.

[Kanzaki et al., 2008] Kanzaki, a., Sawai, Y., Shinohara, M., Hara, T., and Nishio, S. (2008). Quorum-Based Consistency Management for Data Replication in Mobile Ad Hoc Networks. *2008 International Symposium on Applications and the Internet*, pages 357–360.

[Karumanchi et al., 1999] Karumanchi, G., Muralidharan, S., and Prakash, R. (1999). Information dissemination in partitionable mobile ad hoc networks. *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems*.

[Kleerebezem et al., 1997] Kleerebezem, M., Quadri, L. E., Kuipers, O. P., and de Vos, W. M. (1997). Quorum sensing by peptide pheromones and two-component signal-transduction systems in Gram-positive bacteria. *Molecular microbiology*, 24(5):895–904.

[Kleijnen, 2005] Kleijnen, J. P. (2005). An overview of the design and analysis of simulation experiments for sensitivity analysis. *European Journal of Operational Research*, 164(2):287–300.

[Krugman and Wells, 2015] Krugman, P. and Wells, R. (2015). Economics. page 1200.

[Law, 2007] Law, A. M. (2007). *Simulation modeling and analysis*.

[Law, 2014] Law, A. M. (2014). A tutorial on design of experiments for simulation modeling. *Proceedings of the Winter Simulation Conference 2014*, pages 66–80.

[Li and Tian, 2012] Li, Y. H. and Tian, X. (2012). Quorum sensing and bacterial social interactions in biofilms. *Sensors*, 12(3):2519–2538.

[Madigan et al., 2013] Madigan, T. M., Martinko, M. J., Bender, S. K., Buckley, H. D., and Stahl, A. D. (2013). *Brock biology of microorganism*. Number Fourteenth edition.

[Malhotra, 2015] Malhotra, J. (2015). A Survey and Comparative Study of Data Deduplication Techniques. 00(c):0–4.

[Mankiw, 2006] Mankiw, N. G. (2006). Principles of Microeconomics. *Cengage Learning*, page 533.

[Mannes et al., 2012] Mannes, E., Nogueira, M., and Santos, A. (2012). A bio-inspired scheme on quorum systems for reliable services data management in MANETs. In *Proceedings of the 2012 IEEE Network Operations and Management Symposium, NOMS 2012*, pages 278–285.

[Martins et al., 2010] Martins, J. A. P., Correia, S. L. O. B., and Júnior, J. C. (2010). Ant-DYMO: A bio-inspired algorithm for MANETS. In *ICT 2010: 2010 17th International Conference on Telecommunications*, pages 748–754.

[Min et al., 2011] Min, J., Yoon, D., and Won, Y. (2011). Efficient deduplication techniques for modern backup operation. *IEEE Transactions on Computers*, 60(6):824–840.

[Misra, Sudip, Isaac Woungang and Misra., 2009] Misra, Sudip, Isaac Woungang and Misra., S. C. (2009). *Guide to Wireless Ad Hoc Networks*.

[Mohapatra and Krishnamurthy, 2005] Mohapatra, P. and Krishnamurthy, S. V. (2005). *Ad hoc networks: Technologies and protocols*.

[Nealson et al., 1970] Nealson, K. H., Platt, T., and Hastings, J. W. (1970). Cellular control of the synthesis and activity of the bacterial luminescent system. *Journal of bacteriology*, 104(1):313–22.

[Nicholson and Snyder, 2008] Nicholson, W. and Snyder, C. (2008). *Microeconomic theory: basic principles and extensions*.

[Ortiz and Bobadilla, 2003] Ortiz, J. and Bobadilla, L. (2003). Simulación y evaluación de redes ad hoc bajo diferentes modelos de movilidad. *Investigación e Ingeniería*.

[Padmanabhan et al., 2008] Padmanabhan, P., Gruenwald, L., Vallur, A., Atiquzzaman, M., Padmanabhan, P., Gruenwald, · . L., Vallur, A., Atiquzzaman, · . M., and Atiquzzaman, M. (2008). A survey of data replication techniques for mobile ad hoc network databases. *The VLDB Journal*, 17:1143–1164.

[Panait and Luke, 2005] Panait, L. and Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434.

[Parsek and Greenberg, 2005] Parsek, M. R. and Greenberg, E. P. (2005). Sociomicrobiology: The connections between quorum sensing and biofilms. *Trends in Microbiology*, 13(1):27–33.

[Pindyck and Rubinfeld, 2009] Pindyck, R. and Rubinfeld, D. (2009). *Microeconomía*.

[Ramanathan and Redi, 2002] Ramanathan, R. and Redi, J. (2002). A brief overview of ad hoc networks: challenges and directions. *IEEE Communications Magazine*, 40(5):20–22.

[Rebahi et al., 2005] Rebahi, Y., Mujica-V, V. E., and Sisalem, D. (2005). A reputation-based trust mechanism for ad hoc networks.

[Rivest, 1992] Rivest, R. M. L. f. C. S. (1992). The MD5 Message-Digest Algorithm.

[Ross, 2014] Ross, S. M. (2014). *Introduction to probability models*. Academic press.

[Ross-Gillespie and Kümmerli, 2014] Ross-Gillespie, A. and Kümmerli, R. (2014). Collective decision-making in microbes. *Frontiers in Microbiology*, 5(MAR):1–12.

[Rubinstein et al., 2006] Rubinstein, M. G., Moraes, I. M., Campista, M. E. M., Costa, L. H. M. K., and Duarte, O. C. M. B. (2006). A Survey on Wireless Ad Hoc Networks. *Mobile and Wireless Communication Networks*, pages 1–33.

[Russell and Norvig, 2010] Russell, S. J. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*.

[Saltelli, 2004] Saltelli, A. (2004). Global Sensitivity Analysis: An Introduction . *proceedings of the 4th International conference on sensitivity analysis of model output (SAMO 2004)*, (February):27–43.

[Sawai et al., 2006] Sawai, Y., Shinohara, M., and Kanzaki, A. (2006). A Consistency Management Method Based on Quorum Systems with Pointer Information in Ad Hoc Networks. 47:4–8.

[Shinohara et al., 2007] Shinohara, M., Hara, T., and Nishio, S. (2007). Data Replication Considering Power Consumption in Ad Hoc Networks. In *Mobile Data Management, 2007 International Conference on*, pages 118–125.

[Sumpter and Pratt, 2009] Sumpter, D. J. T. and Pratt, S. C. (2009). Quorum responses and consensus decision making. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364(1518):743–53.

[Thomas and Manz, 2017] Thomas, E. and Manz, D. (2017). *Research Methods for Cyber Security*.

[Walter Nicholson and Christopher Snyder, 2008] Walter Nicholson and Christopher Snyder (2008). *Microeconomic Theory Basic Principles and Extensions*.

[Waters and Bassler, 2005] Waters, C. M. and Bassler, B. L. (2005). Quorum sensing: cell-to-cell communication in bacteria. *Annual review of cell and developmental biology*, 21:319–46.

[Wedde and Farooq, 2005] Wedde, H. F. and Farooq, M. (2005). The wisdom of the hive applied to mobile ad-hoc networks. In *Proceedings - 2005 IEEE Swarm Intelligence Symposium, SIS 2005*, volume 2005, pages 351–358.

[Weyns et al., 2005] Weyns, D., Parunak, H., and Michel, F. (2005). Environments for multiagent systems state-of-the-art and research challenges. *Environments for multi-agent systems*, pages 1–47.

[Williams et al., 2007] Williams, P., Winzer, K., Chan, W. C., and Cámara, M. (2007). Look who's talking: communication and quorum sensing in the bacterial world. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 362(1483):1119–1134.

[Wu et al., 2010] Wu, H. J., Peng, L., and Chen, W. W. (2010). The optimization theory of file partition in network storage environment. *Proceedings - 9th International Conference on Grid and Cloud Computing, GCC 2010*, pages 30–33.

[Yin and Cao, 2004] Yin, L. and Cao, G. (2004). Supporting cooperative caching in ad hoc networks. In *Proceedings - IEEE INFOCOM*, volume 4, pages 2537–2547.

[Zhong et al., 2003] Zhong, S., Chen, J., and Richard Yang, Y. (2003). Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 03, pages 1987–1997.