Analysis of Connectivity in Diffusion-Based Molecular Nano Communication Networks

Arash Fereidouni

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
January 2013
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Muhammed Salamah
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Doğu Arifler
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Doğu Arifler

2. Assoc. Prof. Dr. Muhammed Salamah

3. Asst. Prof. Dr. Gürcü Öz

ABSTRACT

A nanonetwork is an interconnection of nano devices that are made up of nano-scale components. Several approaches for designing and implementing nanonetworks have been presented in recent years. Diffusion-based molecular communication is one of these approaches that use molecules as means of transmitting information in network. In diffusion-based molecular communication, molecules or particles diffuse in an aqueous environment under Fick's laws of diffusion to move from transmitter to receiver. In order to have full cooperation among nano devices, there must exist a communication path between every communicating pair. Hence, the primary aim of this study is to employ methods used for analyzing random networks to evaluate connectivity properties of nanonetworks that employ diffusion-based molecular communication techniques. Extensive simulations have been performed to investigate the effects of varying node density, number of particles released per node, and concentration threshold for detection at the nodes. The corresponding results in two and three-dimensional environments have been presented.

Keywords: Connectivity, diffusion, molecular communication, nanonetwork

ÖZ

Nano-ağlar, nano-ölçekte bileşenleri olan nano-aygıtların birbiriyle bağlanmasıyla oluşur. Son yıllarda, nano-ağların tasarım ve oluşturulması için çeşitli yaklaşımlar önerilmiştir. Difüzyona dayalı moleküler iletişim, yani ağda bilginin moleküller kullanarak taşınması, bu yaklaşımlardan bir tanesidir. Difüzyona dayalı moleküler iletişimde moleküller sıvı ortamda Fick'in difuzyon kanunu ile vericiden alıcıya hareket eder. Nano-aygıtlar arasında tam işbirliği için her verici-alıcı çifti arasında bir iletişim yolunun olması gerekir. Dolayısıyla, bu çalışmanın esas amacı, rasgele ağların analizinde kullanılan metodları kullanarak difüzyona dayalı moleküler iletişim kullanan nano-ağların bağlantısallık özelliklerini analiz etmektir. Aygıt sıklığı, aygıt başına yayılan parçacık sayısı, aygıtların yoğunluk algılama eşiği değerlerinin bağlantısallıktaki etkilerini incelemek için simulasyonlar yapılmıştır. Hem iki hem de üç boyutlu ortamlarda elde edilen sonuçlar değerlendirilmiştir.

Anahtar Kelimeler: Bağlantısallık, difüzyona, moleküler iletişim, nano-ağlar

To my family,

for they love and support

ACKNOWLEDGEMENT

First and foremost I offer my sincerest gratitude to my supervisor, Dr. Dogu Arifler, who has supported me throughout my thesis with his patience and knowledge. He truly inspired me during my Master's study as well as introduced me to a novel field in computer science. I attribute the level of my Masters degree to his encouragement and effort and without his guidance I would not be able to overcome all the obstacles in the completion of this research work.

I would also like to thank Dr. Dizem Arifler for her valuable suggestions on preparing my conference presentation for the 4th NaNoNetworking Summit in June 2012.

Last but not least, I would like to express my deepest gratitude to my family, who provided me the opportunity to study and their love and support have always given me energy to accomplish my goals.

TABLE OF CONTENTS

ABSTRACT iii

ÖZ iv

DEDICATION v

ACKNOWLEDGEMENT vi

LIST OF TABLES ix

LIST OF FIGURES x

1 INTRODUCTION 1

1.1 Creating Nano-Machines 1

1.2 Nano-Machine Architecture 3

1.3 Nanonetworks 5

1.4 Applications Of Nanonetworks 6

1.5 Problem Statement 7

2 MOLECULAR COMMUNICATION 9

2.1 Molecular Communication Vs. Telecommunication 9

2.2 Characteristics Of Molecular Communication 12

2.3 Propagation Models In Molecular Communication 14

2.4 Active Transport Vs. Passive Transport 15

2.4.1 Passive Transport-Based Molecular Communication 15

2.4.2 Active Transport-Based Molecular Communication 16

2.5 Categorization Of Molecular Communication Based On Communication Range 16

2.6 General Representation Of Molecular Communication 17

2.7 Mathematics Of Diffusion 26

3 RANDOM GRAPH MODEL OF THE NETWORK 28
3.1 Connectivity Of A Graph 28
3.2 Overview Of Simulation 29
3.3 Environment 31
3.4 Generating Point Locations 31
3.5 Parameters 32
3.6 Building The Graph Model 33
3.7 Measuring Connectivity 37
4 RESULTS 39
4.1 Percent Connectivity As A Function Of *Q/T* 41
4.2 Percent Connectivity As A Function Of *T/Q* 43
4.3 *Q/T* Versus The Number Of Nodes Required To Achieve 95% Connectivity 45
4.4 *T/Q* Versus The Number Of Nodes Required To Achieve 95% Connectivity 47
4.5 *Q/T* Versus The Number Of Nodes Required To Achieve 100% Connectivity 49
4.6 *T/Q* Versus The Number Of Nodes Required To Achieve 100% Connectivity 51
5 CONCLUSION 54
REFERENCES 56
APPENDIX 62
Appendix A: Source Codes For Simulations 63

LIST OF TABLES

Table 1: Telecommunication and molecular communication 12

Table 2: Example for representation of string 101 in time slots by concentration-based encoding 23

Table 3: Example for representation of string 101 in time slots by using different types of molecules 23

LIST OF FIGURES

Figure 1: Approaches for the development of nano-machines 2

Figure 2: Functional architecture mapping between nano-machines of a micro or nano-robot and nano-machines found in cells 5

Figure 3: Shannon model for communication..... 11

Figure 4: Diffusion-based molecular communication with multiple transmitters and multiple receivers 19

Figure 5: Coding techniques: (a) concentration encoding; (b) molecular encoding 20

Figure 6: Molecular multiple-access channel with two TNs simultaneously communicating with a single RN..... 22

Figure 7: Repeaters for molecular communication networks 25

Figure 8: The steps of the simulation..... 30

Figure 9: Random points generated in a 2D environment 32

Figure 10: An example graph and its largest connected component 38

Figure 11: Percent connectivity as a function of Q/T in 2D environment..... 42

Figure 12: Percent connectivity as a function of Q/T in 3D environment..... 43

Figure 13: Percent connectivity as a function of T/Q in 2D environment..... 44

Figure 14: Percent connectivity as a function of T/Q in 3D environment..... 45

Figure 15: Q/T versus the number of nodes required to achieve 95% connectivity in 2D environment 46

Figure 16: Q/T versus the number of nodes required to achieve 95% connectivity in 3D environment 47

Figure 17: T/Q versus the number of nodes required to achieve 95% connectivity in 2D environment..... 48
Figure 18: T/Q versus the number of nodes required to achieve 95% connectivity in 3D environment..... 49
Figure 19: Q/T versus the number of nodes required to achieve 100% connectivity in 2D environment..... 50
Figure 20: Q/T versus the number of nodes required to achieve 100% connectivity in 3D environment..... 51
Figure 21: T/Q versus the number of nodes required to achieve 100% connectivity in 2D environment..... 52
Figure 22: T/Q versus the number of nodes required to achieve 100% connectivity in 3D environment..... 53

Chapter 1

INTRODUCTION

The roots of nanotechnology can be traced back to the “There’s Plenty of Room at the Bottom” speech of the Nobel Prize winner physicist, Richard Feynman in 1959 [1]. The speech was about possibilities of creating tiny but powerful devices that may be employed in a wide range of applications in the future. Nanotechnology is a technology for devices at nanometer scale. It is the study of creating and developing devices and structures having at least one dimension sized from 1 to 100 nanometers. These nano-machines can then be used to construct more complex devices such as nano-robots and nano-processors [2].

1.1 Creating Nano-Machines

Generally, there are three different approaches for creating nano-machines as displayed in Fig. 1: top-down, bottom-up and bio-hybrid [2] [3].

- In the top-down approach, current electronic devices are scaled down from micro to nano level using advanced manufacturing techniques. Quantum effects are the downside of this approach and electrons may have different behavior at nano-scale. This approach is still at an early stage.

- In the bottom-up approach, a bio-inspired approach is used. Using molecules as building elements, molecular components chemically assemble themselves to build nano-machines. Although, there is still no available technology for constructing nano-machines molecule by molecule, this is a promising way of creating nano-machines precisely.

- Bio-hybrid approach is about using biological structures, which act like nano-machines, to create more complex systems or to develop new nano-machines. There are many biological structures in nature, especially in cells, which can be considered as nano-machines. Nano-biosensors, nano-actuators, biological data storing components and control units are some examples of these nano-machines [4].

Fig. 1 shows different systems with two sources of origin, man-made and nature, ranging from nanometers to meters. Although nano-machines can be built using any of these approaches, biological nano-machines, due to their unique characteristics in power consumption and communication can cause new developments for nano-machines [2].

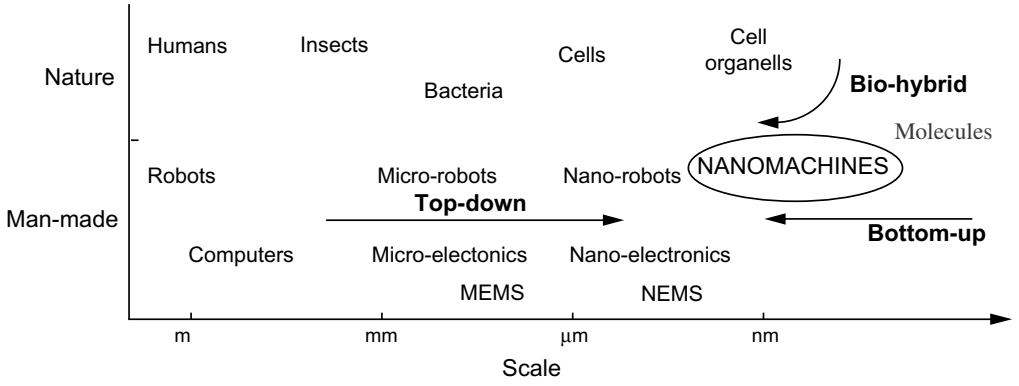


Figure 1: Approaches for the development of nano-machines (reproduced from [2])

1.2 Nano-Machine Architecture

Nano-machines, based on their level of complexity, can have different architectural components. However, the most complete nano-machine can have the following components as further detailed in [2]:

1) *Control unit:* In order to perform the intended task, this unit is responsible for controlling and coordinating other components of nano-machine. It may include a storage section for storing information in the nano-machine.

2) *Communication unit:* By employing transmitters and receivers, this unit is used for communication.

3) *Reproduction unit:* This unit can make each component of the nano-machine by using external sources and then assemble them to make a new nano-machine.

4) *Power unit:* This unit provides the required energy for other components. It can obtain energy from external sources such as light or heat.

5) *Sensor and actuator:* These units are interfaces between the nano-machine and the environment. Their tasks are to gather information about the environment (sensor) and enable the nano-machine to act properly (actuator).

Despite the fact that such complex system cannot be built currently, by looking at biological systems we can find the similar complex nano-machine. For example, living cells are the biological structures that have all the units above and are able to perform tasks that a complex nano-machine is expected to perform. Here is the

mapping between a generic nano-machine and a living cell as discussed in detail in [2]:

1- *Control unit*: The nucleus can be considered as a control unit as it has the ability to understand all the necessary instructions for an intended task.

2- *Communication*: Gap junctions, hormonal and pheromonal receptors on the cell membrane can work as transmitters and receivers in a cell.

3- *Reproduction*: Centrosomes and some molecular motors are involved in the reproduction process.

4- *Power*: Mitochondrion generates most of the chemical substances, which are used as the required energy for cellular processes.

5- *Sensors and actuators*: Various sensors and actuators exist in a cell. Transient Receptor Potential channels for taste and *flagellum* of the bacteria for locomotion are two examples.

More details of these biological structures can be found in [2] [5] [6].

Fig. 2 from [2] illustrates the mapping between a robot nano-machine and a living cell.

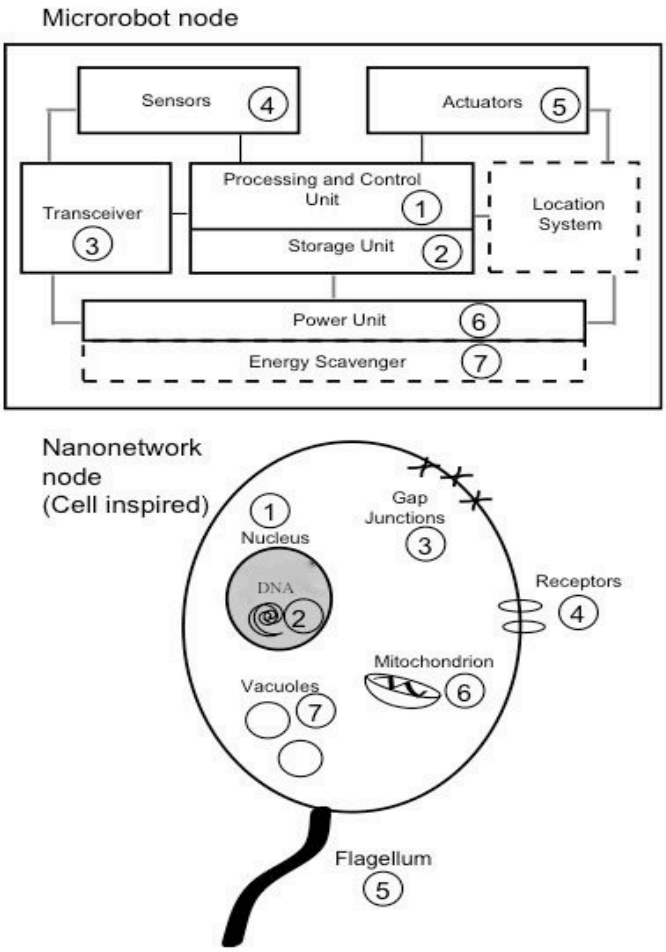


Figure 2: Functional architecture mapping between nano-machines of a micro or nano-robot and nano-machines found in cells (reproduced from [2])

1.3 Nanonetworks

Although nano-machines enable performing tasks at nano-scale, they have disadvantages too. Due to their size and simplicity, nano-machines are capable of performing only simple computation, sensing and actuation tasks. Hence, in order to benefit more from them, we can connect multiple nano-machines in a distributed way to build a nanonetwork that can execute more complex tasks [2].

A nanonetwork expands the capabilities of nano-machines in terms of complexity and range of operation by allowing them to communicate with each other and share data.

1.4 Applications Of Nanonetworks

Nanonetworks can be used in many different applications in various areas such as biomedicine, environment, industry and military.

- *Biomedicine:* The main usage of nanonetworks is in the biomedical field. Since nanonetworks, particularly those using molecular communication, are tiny and biocompatible, they are the best choice for using in intra-body applications where control of system at molecular level is needed. For example, a drug delivery system can transport drug molecules to desired locations [7] [8]. Nano-sensor networks can be used for health monitoring and diagnostic systems [9]. Tissue engineering also can be another possible application exploiting nanonetworks. In [5] other possible applications in biomedical field has been proposed. Lab-on-a-chip applications are useful for diagnosing disease in medical field. In these applications biological samples can be analyzed chemically on a very small chip. Molecular communication can transport molecules through different locations of the chip.

- *Industry:* Nanonetworks can also be employed in industrial applications. They can help with developing new materials and controlling the quality of products. For instance, in food and water quality control, nanonetworks can be helpful. Nano-sensors may detect very small bacteria and toxic materials that cannot be detected by traditional technologies [2]. Consumer goods also can benefit from nanonetworks. As an example, security helmets of motorcycles or race cars can be equipped with

nano-sensors that will affect rider security; or by embedding nanonetworks into vehicles or other machines, we may provide the automation services [10].

- *Military:* Military applications can take advantage of nanonetworks as well. Nano-sensor and nano-actuator networks can be deployed over the battlefield to detect aggressive chemical and biological agents and to respond to them appropriately [11].

- *Environment:* In environmental applications, nanonetworks can be used for biodegradation or air pollution control [12] [13]. In the case of biodegradation, nanonetworks can sense and label materials that can be later located and processed by nano-actuators.

1.5 Problem Statement

In this thesis, different types of nanonetworks will first be introduced. Then, the importance of molecular communication based nanonetworks as a promising method for novel applications will be discussed. This work will consider diffusion-based molecular communication, which will be discussed in detail. Various components and steps of molecular communication will be summarized. Connectivity is a necessary requirement in a network if maximal cooperation among nodes is necessary in a given application. To this end, the nanonetwork will be modeled as a random graph and its connectivity properties will be assessed with respect to node density, number of molecules or particles released per node, and concentration threshold for detection at the nodes in two- and three-dimensional environments. The primary aim will be to evaluate the connectivity of the network by analyzing the connectivity of the graph.

The remainder of the thesis is organized as follows: Chapter 2 introduces molecular communication and its differences from telecommunication. The architecture and different components for this type of communication are explained. Chapter 3 describes the graph model of the network, the method for analyzing the connectivity percentage of the network using the associated graph and parameters that affect connectivity. Chapter 4 shows the results of the simulation by considering different values for node density, number of released particles and detection threshold. Finally, Chapter 5 concludes the thesis and outlines future possible investigation related to this study.

Chapter 2

MOLECULAR COMMUNICATION

2.1 Molecular Communication Vs. Telecommunication

Nanonetworks can be classified according to different methods of communication. The two main alternatives for communication in the nano-scale are based either on electromagnetic communication or on molecular communication [2] [14] [15] [16]. Since classical communication techniques such as electromagnetic communication have several drawbacks at the nano-scale, such as complexity and high power consumption, the use of molecular communication has received more attention of researchers in recent years. Molecular communication is a biocompatible alternative to traditional communication technologies at the nano scale. However, significant research has also been conducted for nano electromagnetic communication techniques [14].

Molecular communication is defined as the transmission and reception of information by means of molecules. In this work, the words “molecules” and “particles” will be used interchangeably. There are several differences between telecommunication and molecular communication. For instance, propagation of electromagnetic waves in space and movement of signals in cables enable communication in wireless and wired telecommunication respectively, which makes this type of communication very fast and reliable. On the other hand, in molecular communication, information transfer is by molecules in an aqueous system and since

the movement of molecules is slow and stochastic, molecular communication is slow and unreliable. The advantage of molecular communication is that molecules can transmit complex data, like a biological function, whereas in traditional communication this is not feasible. Molecular communication has also unique properties like energy efficiency and biocompatibility, which can be very effective in some specific applications. Another important thing to mention is that unlike conventional wireless devices, nano-scale machines are expected to produce energy by themselves instead of getting it from an outside source. Since it may be not practical for them to use an external power, they should provide a mechanism to prepare the sufficient energy to use in their tasks [2].

Both telecommunication and molecular communication are vulnerable to noise existing in the environment. While for traditional networks, noise is overlapping of an undesired signal with information-carrying signals; in molecular communication, there are two different sources for noise. First, as in telecommunication, noise can happen when two molecular signals overlap each other. For example, when two sources release identical messenger molecules, receiver will have problem in decoding information. Noise can also originate when an undesired reaction between information molecules and other molecules in the environment occurs and these reactions change the original message. Thermal energy and electromagnetic fields in the environment also interact with both information molecules and nano-machines and may cause randomness in communication.

In 1948, Claude Shannon proposed a mathematical model for communication. The model, as illustrated in Fig. 3, assumes the following: Information source produces messages and transmitter sends them by converting them to electromagnetic signals.

While the signal is propagating in the channel to reach the designated receiver, it may be affected by noise in the channel. After reception of the signal, it is converted to the message. The communication is successful if the transmitted message and the received message are equal.

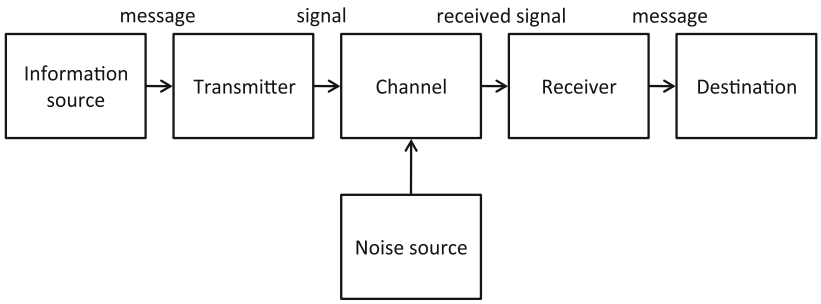


Figure 3: Shannon model for communication

Molecular communication can be described using this model too. While the purpose here is also to transfer a message from transmitter to receiver, the means of communication is different. Again, the information source produces the message, but the transmitter sends it by converting it to signal molecules instead of electromagnetic waves. This conversion can be done by using either the number or the type of molecules. Then, these molecules propagate through the channel that is an aqueous environment. The propagation can occur in two ways: active or passive. Following the Shannon model, in this type of communication also noise in the environment can damage the message, as explained earlier. Finally, after signal molecules hit the designated receiver, they will be converted to the original intended message if possible [17] [2] [15].

Table 1 which is reproduced from [6] shows the comparison between traditional communications and molecular communication in terms of speed, range, etc. [2] [15] [18]. Whereas conventional communications such as electromagnetic waves and electrical signals have higher speed and range, biocompatibility of molecular communication makes it the best choice for use in certain medical applications.

Table 1: Telecommunication and molecular communication (reproduced from [6])

	Telecommunication	Molecular communication
Information carrier	Electromagnetic waves, electrical/optical signals	Chemical signals
Media	Space, cables	Aqueous
Speed	Speed of light (3×10^8 m/s)	Extremely slow (nm$\sim$μm/s)
Range	Long distance ($\sim$km)	Short distance (nm$\sim$m)
Information	Texts, audio, videos	Chemical reactions, states
Other features	Reliable, high energy consumption	Unreliable, biocompatible energy efficient

2.2 Characteristics Of Molecular Communication

Molecular communication consists of several biological components in an aqueous environment. For instance, biological nano-machines may act as sender and receiver or a molecule can act as an information carrier. So, there are some unique characteristics for this type of communication. The following review will be largely based on [5] [6].

Stochastic Communication: The movement of molecules is fundamentally stochastic due to the environmental noise. Communication components also may stochastically react to information molecules or degrade over time. These two facts cause molecular communication to be a stochastic communication. In order to overcome this issue, sender should release large number of molecules to increase the signal to noise ratio. So, if some molecules degrade during propagation, the communication

will not be impacted and non-information molecules (environmental noise) will not be able to trigger chemical reaction at receiver nano-machine [15].

Large Communication Delay: Environmental noise not only makes molecular communication stochastic, but also they cause large delays in propagation of molecules. The speed of propagation is very low in both active and passive transport (i.e., micrometers per second in an aqueous environment). Therefore, the time that molecules are released from sender and the time that they reach to receiver could vary by hours. One important point here is that since some molecules may remain in the environment for a long time, sender should wait until all the molecules from previous communication degrade before starting a new communication. In this way, it prevents old information molecules from interfering with the new one.

Molecule Based Coding: In molecular communication, information can be encoded in different ways by molecules. For example, type of information molecules, three-dimensional structure, chemical structure (e.g., protein), sequence information (e.g., DNA), or concentration (e.g., calcium concentration) may all be used for encoding information [19]. After encoding of information, molecules propagate and hit the receiver. Here, receiver chemically reacts to molecules. The amount of information encoded in the molecule depends on the decoding ability of the receiver.

Biocompatibility: In molecular communication, coding, sending, propagating and receiving are all done by means of information molecules. This way of communicating is similar to what is done by biological systems, which enables nano-machines to contact directly to biological systems and interact with them. Hence, this

considerable feature of molecular communication can make it valuable in medical applications in which nano-machines may be inserted into a biological system.

Energy Efficiency: In comparison with other types of communication, molecular communication is very energy efficient. For instance, under some circumstances, a type of a molecular motor, myosin, converts chemical energy to mechanical work with 90% energy efficiency [6] [20] [21]. The environment must somehow provide the chemical energy. For example, harvesting energy (e.g., glucose) may occur in a human body [15] [22].

2.3 Propagation Models In Molecular Communication

Molecular communication, based on propagation models, can be divided into different categories such as walkway-based, flow-based or diffusion-based communication.

Walkway-based: In walkway-based molecular communication, special substances such as molecular motors are used to enable the propagation of the molecules on particular paths [23] [24].

Flow-based: In flow-based molecular communication, the movement and agitation of molecules are guided by a flowing fluidic medium such as blood or wind and is directed and predictable. Pheromonal communication uses this type of propagation [25].

Diffusion-based: In diffusion-based molecular communication, molecules diffuse spontaneously in a fluidic medium. Their movement and turbulence are under the laws of diffusion so they are completely random and unpredictable. Two well-known

examples using diffusion-based propagation are calcium signaling [2] [23] [26] and quorum sensing among bacteria [27].

2.4 Active Transport Vs. Passive Transport

Molecular communication is a common way of communication between biological systems. Different types of molecule-based communication can be found within and between living cells. These types can be categorized based on how information molecules propagate through the environment. Do they simply diffuse in the medium or they consume energy to direct their propagation? The first approach that doesn't need extra energy for propagation is called passive transport-based and the latter type is active transport-based molecular communication [6].

2.4.1 Passive Transport-Based Molecular Communication

As mentioned above, in passive transport molecules propagate freely in the environment without using any power, using the laws of diffusion. In passive transport, molecules move in all possible directions. So, it is very likely that the time of propagation to the destination for each molecule varies from others, which can cause problems in decoding the information. Also, this type of propagation is very slow, as the time to reach a receiver increases with the square of the distance even in one-dimensional transport. In addition, a large number of molecules are needed for reaching to a distant receiver. For instance in a passive transport-based molecular communication that decoding is done by measuring the concentration of molecules at receiver, some molecules may wander around, getting lost and being destroyed on the way. So, to compensate, the sender should release a large number of molecules. Three examples of passive transport-based molecular communication in biological systems are: Gap junction mediated diffusion-based molecular communication,

reaction diffusion-based molecular communication and free diffusion-based molecular communication [6].

2.4.2 Active Transport-Based Molecular Communication

Unlike passive-based molecular communication, in active mode, molecules propagate directly toward receiver. However to achieve this type of communication, an appropriate infrastructure should be implemented (microtubules, molecular motors, vesicles and filaments). Since molecules in active transport use chemical energy to move in the environment, it is a much faster communication mechanism, it can propagate signal molecules over long distances and because it is a directional communication, the probability of reaching to destination is higher. Thus, there is no necessity of sending large number of molecules whereas the required energy for overcoming chemical interactions between molecules should be supplied. Passive transport is not capable of transporting large molecules and vesicles properly because of their size; however, active transport has enough force to transport large molecules.

Two examples of active transport-based molecular communication are molecular motor-based molecular communication [6] [28] [29] [30] and bacterial motor-based molecular communication [6] [31] [32].

2.5 Categorization Of Molecular Communication Based On Communication Range

Molecular communication can be categorized into three categories based on their effective range of transmission:

Short-range communication: This includes ranges from nanometer to millimeter. In these ranges, techniques like molecular motors and calcium signaling have been proposed [2] [23] [26].

Medium-range communication: This includes ranges from micrometer to millimeter. Flagellated bacteria and catalytic nano-motors are two techniques that have been suggested for this range of communication [24].

Long-range communication: This includes ranges from millimeter to meter. Pheromones are capable of transporting information for this range [2] [16].

The details about this categorization have been discussed in [2] [15] [16] [33].

To summarize the three different classifications of molecular communication, in diffusion-based molecular communication, since the movement of molecules is solely dependent on laws of diffusion (passive transport), molecules can propagate only short distances. So, it can be classified in short-range molecular communications.

In walkway- and flow-based propagation, due to use of extra energy (active transport) and pre-defined paths, molecules can propagate to longer distances including medium- and long-ranges.

2.6 General Representation Of Molecular Communication

A molecular communication system consists of several components. Information molecules that carry information across the network, sender nano-machines that release information molecules, receiver nano-machines that detect information

molecules and the environment in which information molecules propagate from sender nano-machine to receiver nano-machine [2] [34] [6] [5].

In methods that information molecules do not diffuse into the environment, there are also transport molecules, which have the responsibility of transporting information molecules in the environment [35] [5] [2] [16].

Since molecular communication is in an aqueous environment, it deals with significant amount of noise that must be taken into account when designing such systems. The source of this noise can be thermal energy, electrical fields, magnetic fields or other molecules and nano-machines that don't participate in molecular communication, such as water molecules or molecules that prepare that necessary energy for nano-machines.

The general procedure in diffusion-based molecular communication is the following: First, information is encoded into information molecules. Then sender releases the information molecules into the environment. These molecules diffuse through the environment from sender to receiver. Finally, receiver takes the information molecules and decodes them into a chemical reaction at the receiver nano-machine. The process can be divided into different steps as follows (see also Fig. 4):

- Encoding
- Sending
- Propagating
- Receiving
- Decoding

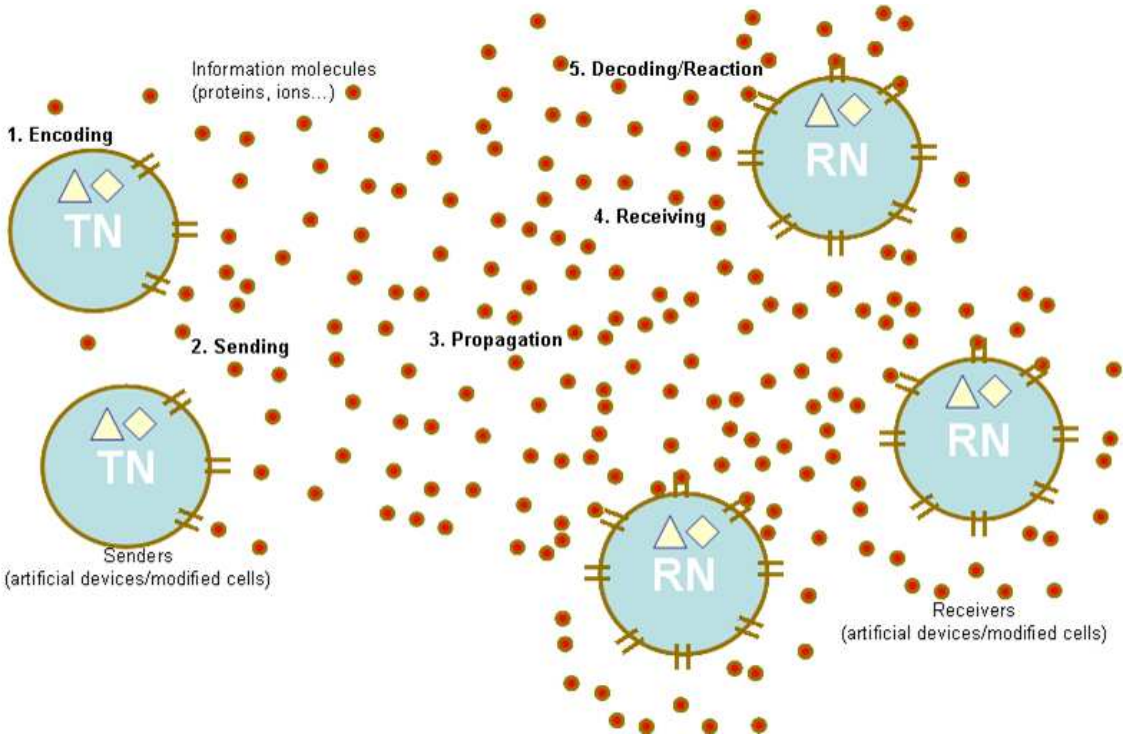


Figure 4: Diffusion-based molecular communication with multiple transmitters and multiple receivers (reproduced from [36])

Encoding: Encoding is the process in which a sender nano-machine translates data into information molecules. This translation can be done by different methods.

a) First method is to use different types of molecules for each bit of information. For example molecule type *A* may represent a 0 and molecule type *B* may represent a 1 in a binary string of information.

b) The other method is using the concentration/number of molecules as a criterion of recognizing different bits [37]. For example, if the number of released molecules by sender is above *N* molecules, then it is considered as 1, otherwise if it is less than *N* molecules it is considered as 0.

Fig. 5 shows two different mentioned methods for encoding information.

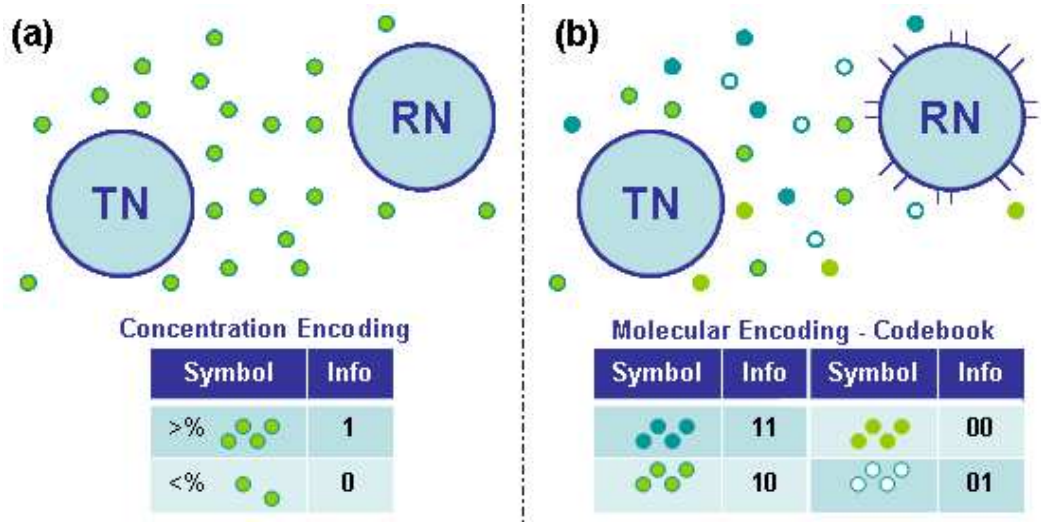


Figure 5: Coding techniques: (a) Concentration encoding; (b) Molecular encoding (reproduced from [36])

The amount of information encoded by the sender depends on the structure of receiver nano-machine [16] [38]. It means that the receiver should be capable of decoding the amount of information relating to the number of possible configurations [6]. These configurations, which are called *after states,* have been discussed comprehensively in [38]. They enable the receiver to choose one state amongst others. For example, according to [6], if a receiver nano-machine handles 2 configurations, it can decode only 1 bit of information at a time or in order to decode 2 bits of information simultaneously, receiver needs to be capable of handling 4 configurations. In other words, if N= Number of bits that can be decoded in same time, then:

$$No.of\ possible\ configuration\ in\ receiver = 2^N$$

The information should be transmitted between the transmitter and receiver through the propagation of certain molecules [39]. These molecules are called messenger molecules and they can be chosen from special type of molecular structures such as protein, peptide, DNA sequence, etc. [34] [15]. These molecules have to be a shield for information molecules and protect them during the transportation. They must also have some properties to be appropriate for this type of communication. They must be biocompatible and not toxic to components of molecular communication system. In addition their building blocks should be accessible in the environment so transmitter can build them easily during communication.

Sending: Sending phase happens when sender nano-machine releases information molecules into the environment. It can either open a gate so molecules can diffuse away or by a chemical reaction it produce transport molecules [15]. Since nano-machines are very small and low capacity devices, a sender nano-machine may handle limited number of information molecules and energy. So, there should be a mechanism that sender can supply chemical energy and molecules from environment.

Also, as depicted in Fig. 6, in order to increase the number of emitting molecules resulting in having more reliable communication, multiple sender nano-machines can be used. If they release the same information molecules at the same time, the signal in the environment would be stronger and easier to detect by receivers [37].

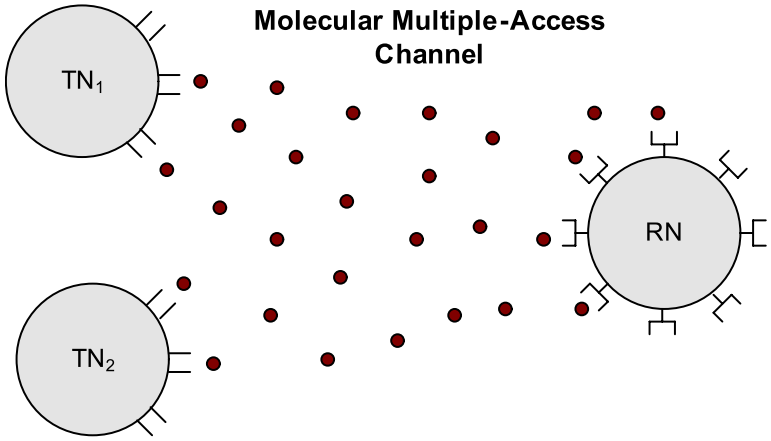


Figure 6: Molecular multiple-access channel with two TNs simultaneously communicating with a single RN (reproduced from [40])

Besides, in this system, information, which is a sequence of symbols, is sent over time. To manage this transportation, time should be divided into different slots and each symbol can be sent in one time slot. This is similar to Time Division Multiple Access (TDMA) mechanism in conventional communication systems, except here, the information is molecules. These symbols can be multiple numbers of molecules if concentration-based technique is used for encoding information, or can be a specific type of molecule if different types of molecules are used to represent information [16]. In first case for example we have time slots *T1, T2, T3*, … and we want to transmit 101. The threshold for deciding whether the number of molecules is representing 0 or 1 has been defined earlier. We suppose that this threshold is 50. So if number of released particles exceeds 50 it represents 1 and if it is less than that it is interpreted as 0. Now all we need to do is to encode the information into particles and send them in 3 time slots as in Table 2:

Table 2: Example for representation of string 101 in time slots by concentration-based encoding

Time Slot	T1	T2	T3
Number of released particles	60	20	70

In second case where several types of molecules are used to represent information string, we can assume that we have 2 types of molecules: Type *A* for indicating a 1 and Type *B* for indicating a 0. Now for sending the above string, 101, we have to do put our symbols in their time slots as in Table 3:

Table 3: Example for representation of string 101 in time slots by using different types of molecules

Time Slot	T1	T2	T3
Type of released molecule	A	B	A

Propagation: Propagation is when released molecules move through the environment to reach the desired destination. This movement can be passive like diffusion [41] or done by transport molecules [35]. In diffusion, signal molecules move randomly according to forces in the medium, so the size of the molecules and viscosity of environment can affect the speed of transmission. Since in this kind of propagation no extra energy is consumed, diffusion is a slow and stochastic transport. Unlike diffusion, in active method, transport molecules consume chemical energy to carry information molecules from sender to receiver. Hence the speed of transmission as well as the likelihood of molecules successfully getting to the desired destination increases.

Because of noise and special characteristics of the aqueous environment, molecules may be damaged or dissolve during the propagation [2] [42]. So an interface molecule may also be necessary to protect information molecules from existing noise in the environment. For instance, a vesicle-based molecule can be used as an interface for information molecules [35]. Placing information molecules in a vesicle-based molecule provides a safe propagation through the environment.

Another problem that may occur is that since each symbol is sent in a time slot it should be received in a specific time slot as well so it has a limited time to propagate and hit the receiver. Sometimes due to the characteristics of the environment and stochastic properties of diffusion-based communication, particles cannot propagate in a desired time and they may have delay reaching the destination. Those particles that do not hit destinations in a proper time can cause problems with the decoding process of next symbols. So, there should be a mechanism that prevents the interference of the decoding process of next symbols. In other words, a lifetime proportional to required time for propagating from sender to receiver should be defined for propagating particles.

One solution for helping particles to enhance their propagation through the environment is to use relay nodes across the network [43]. Since the signal molecules attenuate over distance, these nodes, known as intermediate repeaters, can be placed between sender and receiver nano-machines to amplify the signal to reach to receiver. Fig. 7 shows the basic idea in which two repeater nano-machines (R_p) are placed between the transmitter nano-machine (T_x) and the receiver nano-machine

(R_x). T_x transmits a number of signal molecules, which are detected and amplified by the first R_p and then second R_p to reach R_x.

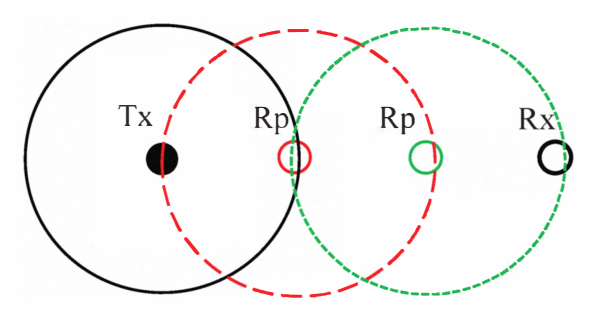


Figure 7: Repeaters for molecular communication networks (reproduced from [43])

Receiving/decoding: The final part of communication is receiving information molecules and decoding them to corresponding actions, which are done by receiver nano-machines. Capturing molecules can either be done by receptors located on the receiver nano-machine or by using special gates [2] [16], like in sender that let the information molecules enter receiver nano-machine.

Receiver then decodes the information molecules to chemical reactions. Chemical reactions may include producing new molecules, performing a task or sending other information molecules. This phase, like encoding, can also use two methods for decoding the information.

a) Concentration-based decoding, which decodes received particles by measuring their concentration and decide what symbol they represent.

b) Another approach is molecule-based decoding in which each molecule represent a specific symbol. It should be noted that in a molecular communication system, transmitter and receiver should use the same encoding/decoding technique; either both use concentration-based or both use molecule-based encoding.

A common problem in concentration-based decoding is the effects of ISI (Inter Symbol Interference) [42]. ISI happens when some messenger molecules from previous time slot have delay and cannot reach the receiver in their intended time. These molecules may reach the receiver in the next time slot that can cause problem in decoding the signal of that time slot.

2.7 Mathematics Of Diffusion

In this thesis, diffusion-based molecular communication is considered. Therefore, a mathematical model of diffusion will be described. According to Fick's laws of diffusion, particles move from regions of higher concentration to regions of lower concentration. Assume that the concentration of particles used in communication is much lower than that of the fluid medium. Fick's first law describes the relationship between the flux and concentration. In the x-direction, we have:

$$\Phi\,(x,t) = -D\,\frac{\partial\, C(x,t)}{\partial\, x} \tag{2.1}$$

where $\Phi\,(x,t)$ is the diffusion flux per unit area per unit time, D is the diffusion coefficient in dimensions of $[length^2\, time^{-1}]$, $C(x,t)$ is the concentration in dimensions of [amount of substance per unit volume], x is the position and t is the time. D is proportional to the squared velocity of the diffusing particles, which depends on the temperature, viscosity of the fluid and the size of the particles. For

biological molecules the diffusion coefficients normally range from 10^{-11} to 10^{-10} m^2/s.

Fick's second law predicts how diffusion causes the concentration to change with time:

$$\frac{\partial\, C(x,t)}{\partial\, t} = D\, \frac{\partial^2\, C(x,t)}{\partial\, x^2} \tag{2.2}$$

The same relationships hold in both y- and z-directions as well. In general, one writes:

$$\overrightarrow{\Phi}(x,y,z,t) = -D\, \nabla\, C(x,y,z,t) \tag{2.3}$$

and

$$\frac{\partial\, C(x,y,z,t)}{\partial\, t} = D\, \nabla^2 C(x,y,z,t) \tag{2.4}$$

where

$\nabla = \left(\frac{\partial}{\partial\, x}, \frac{\partial}{\partial\, y}, \frac{\partial}{\partial\, z}\right)$ and $\nabla^2 = \left(\frac{\partial^2}{\partial\, x^2}, \frac{\partial^2}{\partial\, y^2}, \frac{\partial^2}{\partial\, z^2}\right)$.

Chapter 3

RANDOM GRAPH MODEL OF THE NETWORK

The previous chapter mainly discussed molecular communication between a sender and a receiver. In particular, diffusion-based molecular communication is identified as a basic mechanism that can be used to connect two nano-machines. There is relatively less work on macroscopic properties of nodes as a network that uses the aforementioned communication mechanism [44]. A significant property, connectivity of diffusion-based nanonetworks, is analyzed in this chapter. Similarities to traditional wireless network settings will also be pointed out.

3.1 Connectivity Of A Graph

Connectivity is one of the basic and important concepts in graph theory [45]. It describes the whole situation of a graph, whether the elements of graph are connected to each other or not. In graph theory, a graph is defined by vertices and edges. In undirected graphs, two vertices are connected to each other if there is an edge between them. Otherwise, if there is not any edge between two vertices, there is no connection between corresponding vertices.

Generally, a connected graph is the one in which there is at least one path between every vertex to other vertices and if in one graph there is a vertex that has no connection to other vertices, that graph is not connected.

Nodes and links in a communication network can be modeled as a graph. Assuming each node of network as a vertex and communication link between two nodes as an edge between vertices, one can create a corresponding graph of a network. These nodes can be any computational devices such as computers, sensors, transceivers or nano-machines. The connectivity of a graph illustrates the communication situation of the network. Questions such as “which nodes in the network are connected to each other?”, “which nodes are isolated” or “how many connected components are there in the network?” can be answered by analyzing the corresponding graph model. It is obvious that the connectivity of a graph is an important measure of its robustness as a network.

Connectivity is a crucial factor in both traditional wireless and nano sensor networks. In these types of networks, multiple nodes are distributed through the desired environment. Each node has tasks of sensing, collecting and sending information to a server. The server then performs an action based on the received information. Now, if nodes are not completely connected to each other, in other words there is not 100% connectivity in the network, the information received by the server cannot represent the real situation of the network. Hence, the action by the server may not be appropriate for the network. Therefore, connectivity plays a crucial role not only in such networks but also in applications where maximal cooperation among nodes is necessary.

3.2 Overview Of Simulation

In this work, the connectivity properties of nano-networks that employ diffusion-based molecular communication techniques in a binary channel are analyzed. To this end, a nanonetwork whose nodes are deployed randomly has been modeled as a

random graph [44] [46] and its connectivity properties with respect to node density, number of particles released per node, and concentration threshold for detection at the nodes in two-dimensional (2D) as well as three-dimensional (3D) environments have been assessed. The simulation is divided into 7 steps as illustrated in Fig. 8:

1- Creating a desired 2D or 3D environment

2- Generating random points in that environment, representing our nodes in network

3- Calculation of the distance between each node and all other nodes

4- Assigning values for number of released particles at sender and detection threshold at receiver

5- Deciding which nodes are connected to each other and building corresponding graph

6- Measuring connectivity based on different properties of sender and receiver

7- Evaluating connectivity for different scenarios

Figure 8: The steps of the simulation

MATLAB in conjunction with the Bioinformatics Toolbox is used for implementation of the simulations and processing of results.

3.3 Environment

This study considers both two-dimensional and three-dimensional regions. In order to compare results, a fixed area has been chosen in two dimensions and a fixed volume in three dimensions.

3.4 Generating Point Locations

Modeling networks as graphs can be very useful. To do so, nodes in a nanonetwork are modeled as points in an environment. In this study points are distributed in the environment completely randomly. For two-dimensional areas, the locations of N tiny nodes in a given area $A \subset R^2$ are assumed to be described by a binomial point process $\Psi_{A\,(N)}$ [47]. The binomial point process is a spatial stochastic point process in which the occurrence of any point is equally likely in the region (see, for example, Fig. 9). To justify this, we assume that our points have been distributed completely randomly in the environment. Now if we suppose a sub-region $B, B \subset A,$ and let $\Psi_{A\,(N)}(B)$ specify the number of points of $\Psi_{A\,(N)}$ falling in a sub-region B where $\Psi_{A\,(N)}\,(A) = N$, $\Psi_{A\,(N)}(B)$ has a binomial distribution with parameters N and $P = {|B|}/{|A|}$, where $|A|$ and $|B|$ denote the areas of $|A|$ and $|B|$, respectively.

As a result, the probability that a point falls in a sub-region is directly proportional to the area of sub region. As the area B becomes bigger, the probability that a random point falls in it will be higher and by reducing the size, the probability also decreases.

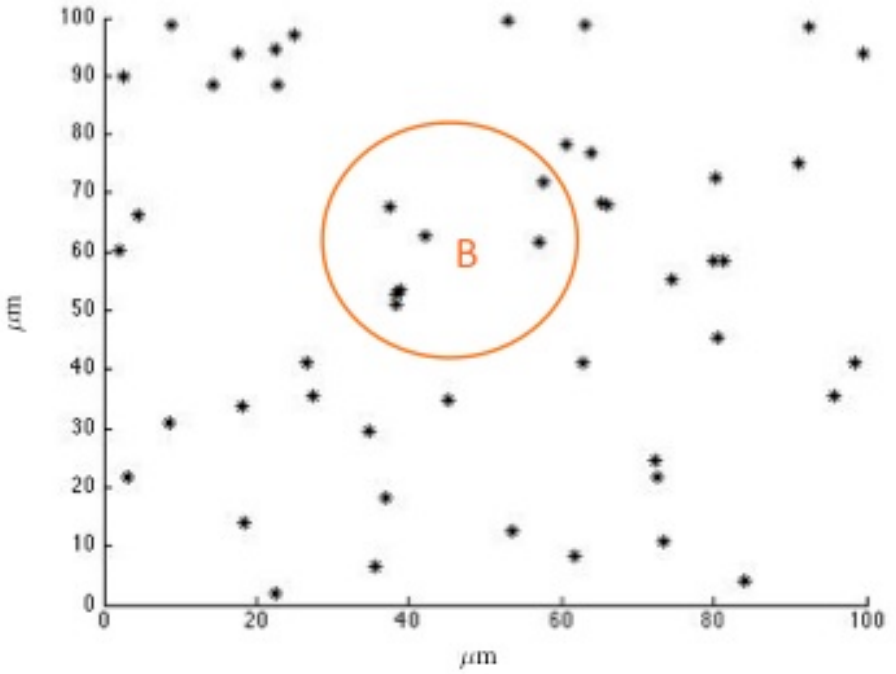


Figure 9: Random points generated in a 2D environment

3.5 Parameters

Number of Particles (Q): In diffusion-based molecular communication, if sender *A* wants to send information to receiver *B*, it releases particles, which diffuse in the environment until they hit the receiver. The number of particles released by sender is an important factor for having connection between *A* and *B*. Since, some particles may get lost in the environment or doesn't hit the receiver in appropriate time due to random walk in the environment, releasing higher number of particles can help having a robust connection. It must also be pointed out that in most systems the number of particles is limited. Also, if number of released particles becomes much more than necessary, it can result in noise for later communication, because some of them may remain in the environment and affect the next signals. Therefore, choosing an appropriate number of particles to release is essential to achieve high percentage

of connectivity in network when there is a limitation for the number of particles and interference constraints.

Detection Threshold (T): At receiver side, receivers decode the received particles according to a pre-defined threshold value. For example by sensing 10 or more particles in 1 μm^3, receiver assumes the sent information has been 1 and by sensing less than 10 in 1 μm^3, it interprets received particles as 0. In this case the detection threshold is 10 particles per 1 μm^3. Now, if the threshold value changes to 20 particles per 1 μm^3, higher number of received particles is needed for receiver to decode 1. Setting this value to higher numbers can help receiver to neutralize the effects of noise in the environment, while it may result in not decoding the information properly since desired number of particles may not be able to hit the receiver in right time. Therefore, detection threshold value also needs to be chosen wisely as it has a significant effect on connectivity.

In this study, various scenarios have been tested and evaluated for different values of detection threshold, different numbers of released particles and multiple numbers of nodes.

3.6 Building The Graph Model

By solving Fick's laws of diffusion, discussed in previous chapter, the propagation of particles can be modeled analytically. If it is assumed that each node releases Q information-carrying particles for an infinitesimal amount of time at time $t = 0$, this process is often referred to as an instantaneous or a "puff" emission. The particles move by free diffusion and in two-dimensional area their concentration as a function

of distance r from the source of emission and the number of emitted particles Q at time $t > 0$ is the solution to Fick's second law in Eq. (2.2) and is given by:

$$C(r,t) = \frac{Q}{4\pi Dt} e^{\frac{-r^2}{4Dt}} \tag{3.1}$$

where D is the diffusion coefficient. It is assumed that the particles are labeled so that their origin can be identified; these labels may be considered as source addresses associated with the nodes. Labels are short molecular chains that are added to particles through bio-molecular engineering techniques. For instance sender i uses label i in its particles and sender j uses label j. When they both release their particles, these particles diffuse in the environment until they hit their receivers. In this case receivers can distinguish between different received particles.

A node i can communicate with node j at a distance r away if the concentration of particles with labels i received by node j exceeds a pre-specified concentration threshold T. In a 2D environment, the maximum concentration level is reached at node j at time $T = \frac{r^2}{4D}$ and is given by:

$$C_{max} = \frac{Q}{\pi e r^2} \tag{3.2}$$

Since all nodes emit Q particles, we can say that nodes i and j, which are separated by a distance r, are connected if $C_{max} > T$; i.e., in this work, the links are bidirectional. In other words, if there is a connection from i to j, there should be a

connection from *j* to *i*. In this model, there is an edge between the source node and the destination node if they can communicate [44].

After nodes are distributed in the network, the maximum concentration at destination nodes from any other nodes is calculated and if the result is above the pre-defined threshold value, an edge between those two nodes will be created. It should be noted that in this study all nodes are both transmitter and receivers. In other words it could be the case that while a node is transmitting particles to some other nodes, its receptors may receive other particles from environment. So, if there are *N* nodes in the network, in a corresponding graph, each node can have 0 up to *n*-1 edges to other nodes. Each edge between two nodes is interpreted as a communication path between them. Thus, a network is connected if a communication path can be established between any two nodes.

The formulas stated earlier are for two-dimensional environments. However, for three-dimensional environments, all the procedures of generating random points as nodes and measuring distance between them are similar to 2D except that the concentration follows a different formula. In 3D, the concentration as a function of distance *r* from the source of emission and the number of emitted particles *Q* at time $t > 0$ from Fick's second law in Eq. (2.2) is given by:

$$C(r,t) = \frac{Q}{(4\pi Dt)^{\frac{3}{2}}}\, e^{-\frac{r^2}{4Dt}} \tag{3.3}$$

Similar to the 2D scenario, node *i* can communicate with node *j* at a distance *r* away if the concentration of particles with labels *i* received by node *j* exceeds a

pre-specified concentration threshold *T*. In a 3D environment, the maximum concentration level is reached at node *j* at time $T = \frac{r^2}{6D}$ and is given by:

$$C_{max} = \left(\frac{3}{2\pi e}\right)^{\frac{3}{2}} \frac{Q}{r^3} \tag{3.4}$$

After the concentration threshold for each node from all other nodes has been calculated, by comparing this value and pre-defined threshold, the communication paths will be created and the resulting graph shows the connectivity situation of the network.

Diffusion-based molecular communication has a lot of similarities with traditional wireless communications. Well-studied connectivity analyses in the context of wireless communication are also applicable in molecular communication with appropriate interpretations. In wireless communications, transmitted signals are subject to a power law attenuation function $\ell(r)$ where *r* is the distance from the signal transmitter. In free space, for *r* larger than the reference distance for antenna far field, $\ell(r) \sim 1/r^2$. Therefore, in particular, free diffusion in 2D is analogous to free space wireless propagation as can be observed from Eq. (3.2): It can be seen that in 2D molecular communication, the maximum concentration value (analogous to signal power) that is used for signal detection is inversely proportional to r^2 as well. In 3D settings, the maximum concentration value is inversely proportional to r^3. This case may correspond to environments with higher path loss exponents. It will be shown in Chapter 4 that it is much harder to achieve connectivity in 3D than in 2D using the same number of particles.

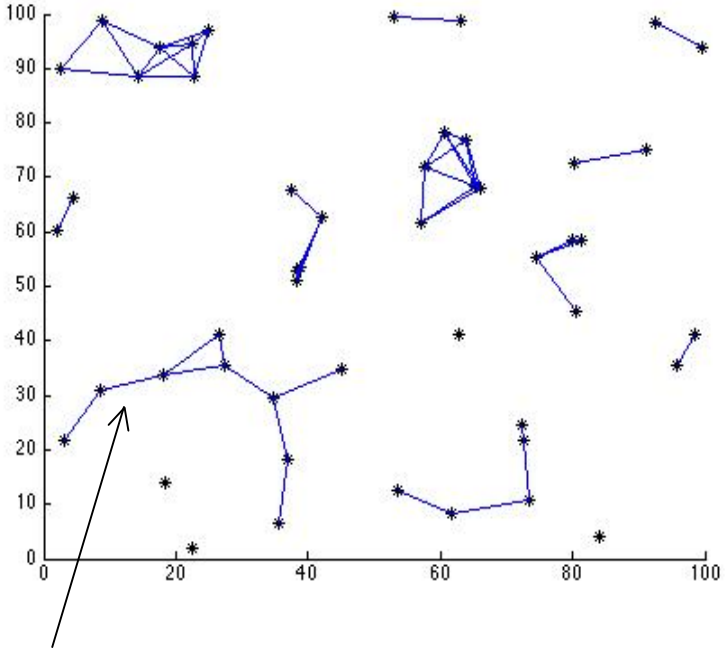
3.7 Measuring Connectivity

Connectivity of the network is evaluated by analyzing the connectivity of the graph. To find connected components of the graph, depth-first search (DFS) algorithm has been used. This algorithm selects a node as the root and explores as far as possible along each branch before backtracking. In order to measure the connectivity percentage of a graph, first the largest connected component of that graph should be found. A graph may consist of several components. The largest one is the one in which higher number of nodes are connected to each other. After comparing number of nodes in each component, the largest component will be selected. Then the number of involved nodes will be calculated and by dividing that number by the total number of nodes in the network, the connectivity percentage of the graph will be obtained. The network is θ-connected if the fraction of nodes in the largest connected component of the network is θ.

$$\theta = \%\, connectivity = \frac{number\ of\ nodes\ in\ largest\ component}{total\ number\ of\ nodes\ in\ network} \times 100\ \% \qquad (3.5)$$

As an example, Fig. 10 shows a graph with 50 nodes that are randomly placed in an area and multiple components. The largest component has been indicated in the figure to evaluate the connectivity percentage of the graph. The number of nodes in the largest component, which is 9, should be divided by 50, the total number of nodes, which results in 18% of connectivity.

$$\theta = \%\, connectivity = \frac{9}{50} \times 100\ \% = 18\%$$



The largest component

Figure 10: An example graph and its largest connected component

Chapter 4

RESULTS

With certain number of released particles and detection threshold, the decision will be made that whether a path should be created between two nodes or not. This measurement has been done with many different numbers of particles and detection thresholds by running simulations. For each pair of values, a graph has been built, edges have been created and the corresponding connectivity has been measured.

In preliminary investigations, by fixing the detection threshold and increasing the number of particles related connectivity was measured. This assessment was done with different fixed detection thresholds and increasing number of particles. After several scenarios, the following observation made: The connectivity was the same when values of number of particles (Q) and detection threshold (T) were different but the ratio of Q/T was the same. In other words, no matter how many particles the sender released and to which value the detection threshold was set, as long as their ratio remained the same, the network had the same connectivity percentage. As an example with Q=100 particles and T=1 particle per 1 μm^3, the resulting connectivity in the network was the same as Q=1000 and T=10 as the Q/T ratio was 100 in both cases.

Likewise, investigations were carried out with different fixed number of particles and decreasing detection threshold (since reducing the detection threshold results in

higher chance of connection). Again various scenarios were tested and it was realized that with different number of released particles (Q) and detection threshold (T), but with the same T/Q ratio, the same connectivity percentage would be achieved. For instance by having Q=10 particles and T=1/100 particle per 1 μm^3, the same connectivity percentage would be found for the network as with Q=100 particles and T=1/10 particle per 1 μm^3. Results can be classified in different categories as follows:

1- Percent connectivity as a function of Q/T, or the ratio of the number of particles released to the detection threshold
(Q variable, T=100 particles/ μm^2 for 2D and T=100 particles/ μm^3 for 3D)
2- Percent connectivity as a function of T/Q, or the ratio of the detection threshold to the number of particles released (T variable, Q=10,000 particles)
3- Q/T, or the ratio of the number of particles released to the detection threshold versus the number of nodes required to achieve 95% connectivity
(Q variable, T=100 particles/ μm^2 for 2D and T=100 particles/ μm^3 for 3D)
4- T/Q, or the ratio of the detection threshold to the number of particles released versus the number of nodes required to achieve 95% connectivity
(T variable, Q=10,000 particles)
5- Q/T, or the ratio of the number of particles released to the detection threshold versus the number of nodes required to achieve 100% or full connectivity
(Q variable, T=100 particles/ μm^2 for 2D and T=100 particles/ μm^3 for 3D)
6- T/Q, or the ratio of the detection threshold to the number of particles released versus the number of nodes required to achieve 100% or full connectivity
(T variable, Q=10,000 particles)

All simulations have been tested in both two and three-dimensional environments. By fixing the size of the environment, comparison between results could be done. For two-dimensional areas 100μm × 100μm area and for three-dimensional volume 100μm × 100μm × 100μm volumes were employed.

4.1 Percent Connectivity As A Function Of *Q/T*

In this simulation the connectivity of network has been demonstrated as a function of *Q/T*. Here, *N* represents the number of nodes in the environment. Different cases shown correspond to *N*=100,200,300,400, and 500. First, the two-dimensional area case is considered. As the Fig. 11 shows, with *Q/T*= 100 μm^2 the connectivity in the network is below 5%. This is true for *N*=100 to 500. As the *Q/T* increases, higher connectivity can be achieved. It is obvious that with certain ratio, increasing the number of nodes equals to higher connectivity in the network. This is because if number of nodes increases in a fixed size environment, the distance between them will be less. Thus it is easier for particles to reach their desired destination. In other words the probability of hitting receiver is higher so the connectivity between nodes also would be higher.

Moreover the connectivity percentage is higher in higher number of nodes; as there is a sudden jump at *Q/T*=200 μm^2 for *N*=500 and at about *Q/T*=300 for *N*=400 and 300. This phase transition is a prevalent behavior in graphs and is consistent with the percolation theory [46]. By having *Q/T*= 350 for 400 and 500 nodes and *Q/T*=550 for 300 nodes we can have a fully connected network in the predefined area; while with 100 and 200, *Q/T* ratio must be above 1000.

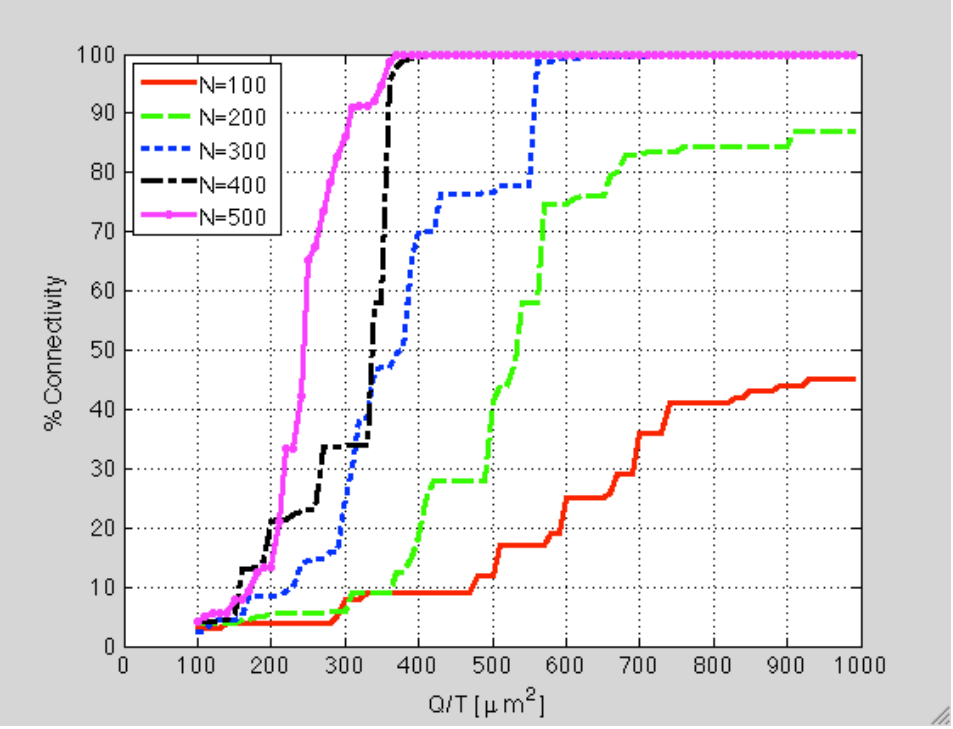


Figure 11: Percent connectivity as a function of *Q/T* in 2D environment

In three-dimensional regions, the results are roughly similar but since the environment is larger and has more degrees of freedom, higher *Q/T* is needed in the network. Hence, according to Fig. 12, for example with 500 nodes, *Q/T* have to be about 60000 μm^3 to have a fully connected network. In other words, for fixed *T*, one needs higher *Q* to achieve the same percent connectivity in 3D. In 3D environments, phase transition behavior can correspondingly be observed.

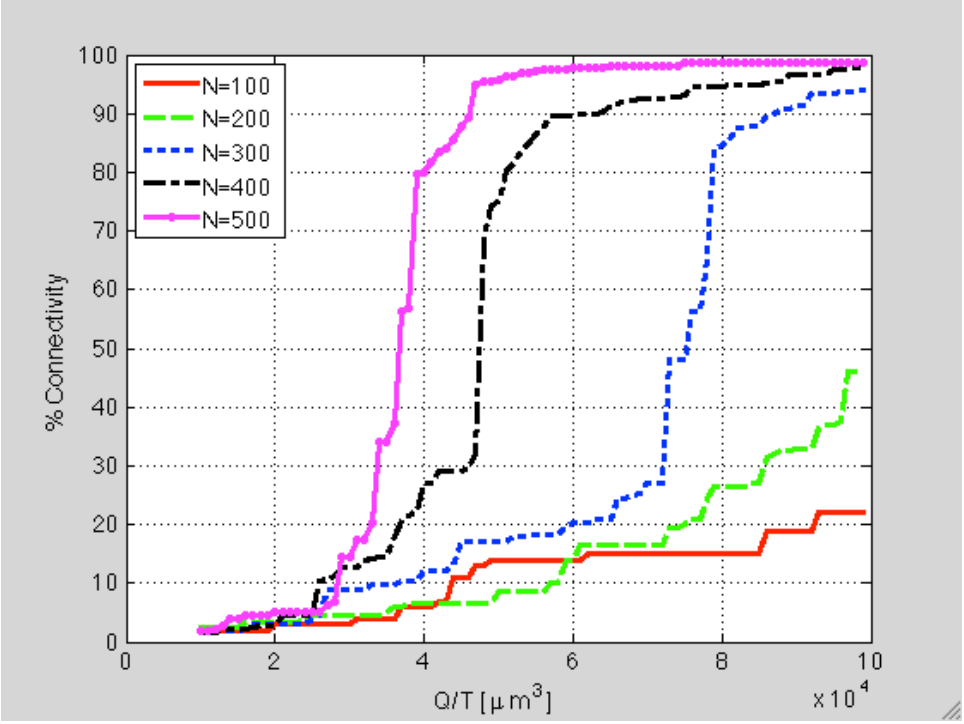


Figure 12: Percent connectivity as a function of *Q/T* in 3D environment

4.2 Percent Connectivity As A Function Of *T/Q*

In this case the connectivity of network has been demonstrated as a function of *T/Q*. Again, different cases are shown correspond to 100,200,300,400 and 500 nodes. As it can be observed from Fig. 13, the connectivity percentage is inversely related to the *T/Q* ratio. In other words, by reducing the detection threshold over number of emitted particles proportion higher percent connectivity can be attained. For example when *T/Q*= 0.01 μm^{-2}, connectivity is less than 5% while we have a connected graph with 0.003, 0.002, 0.0015, 0.001 and 0.0005 μm^{-2} for 500, 400, 300, 200 and 100 nodes respectively. Here again with definite value for *T/Q* and by increasing number of nodes, the distance between them decreases and hitting probability rises. Phase transition also can be seen in both figures in 2D and 3D environments.

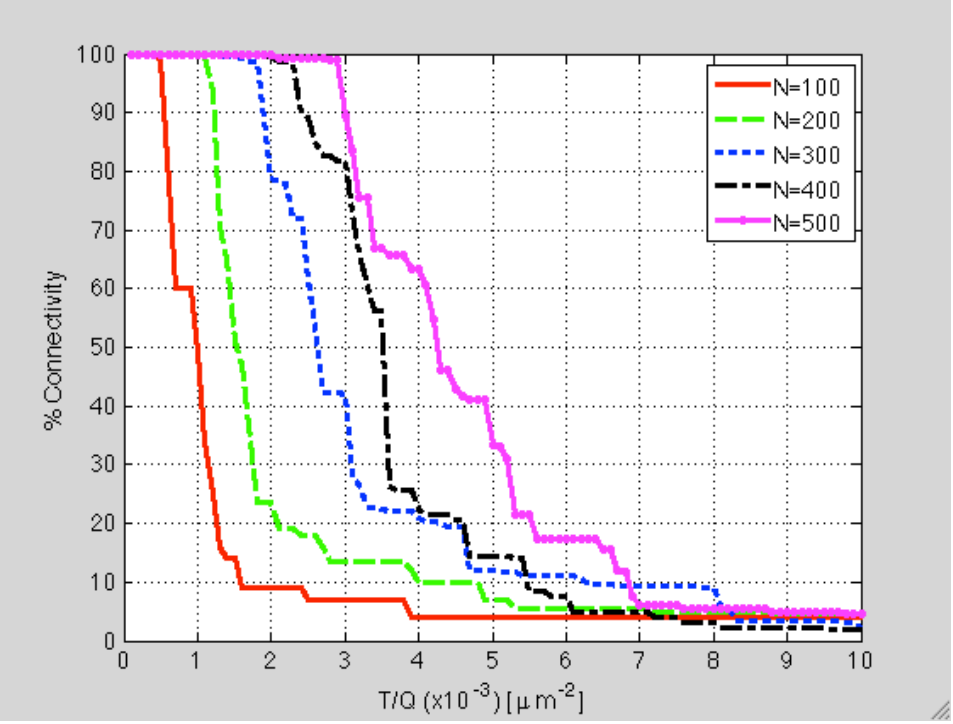


Figure 13: Percent connectivity as a function of *T/Q* in 2D environment

According to Fig. 14, the results for three-dimensional environment are quite similar to 2D. The difference is the value of *T/Q* ratio that achieves high connectivity. For fixed *Q*, one needs lower *T* to achieve the same percent connectivity in 3D.

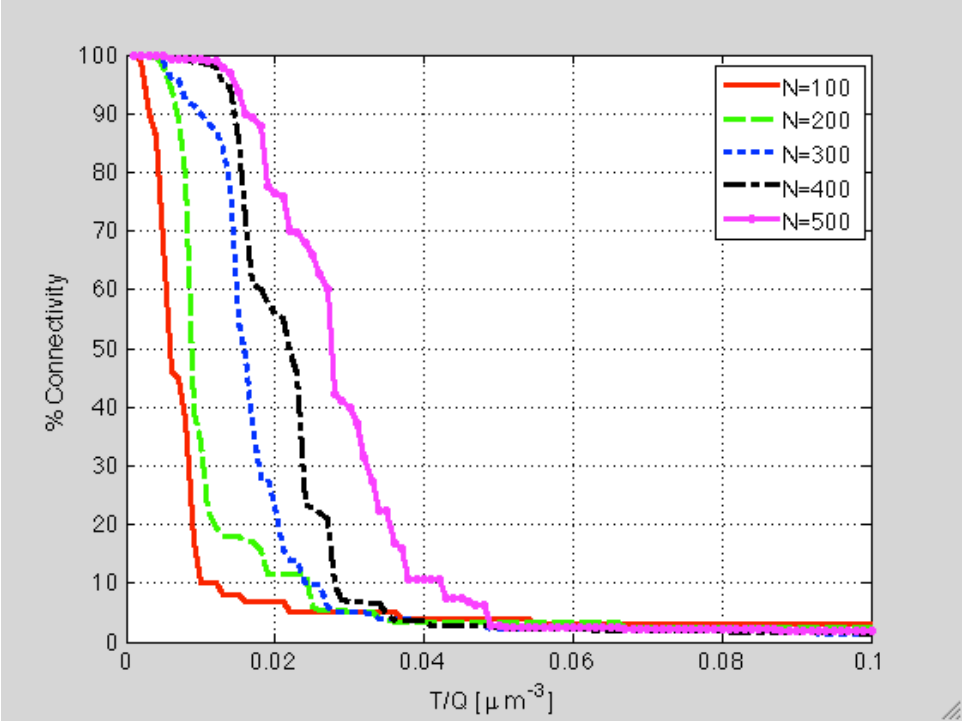


Figure 14: Percent connectivity as a function of *T/Q* in 3D environment

4.3 *Q/T* Versus The Number Of Nodes Required To Achieve 95% Connectivity

Next, the number of particles that must be released per node and the detection threshold that must be employed to achieve 95% connectivity has been analyzed. This has been done by both measuring *Q/T* and *T/Q*.

Since 100% is a strict requirement, in most applications, 95% connectivity may be acceptable. So, in this simulation the *Q/T* has been increased until achieving 95% connectivity and the corresponding ratio has been illustrated in Fig. 15. This has been done with different number of nodes ranging between 100 and 1000.

The ratios reported are calculated based on 10 simulation runs. The standard error bars are also shown in the figures. These results indicate that, for 95% connectivity, *Q/T* decreases nonlinearly with the number of nodes. In two dimensions, with 100 nodes, the proportion of number of emitted particles over detection threshold required for 95% connectivity is 1400 μm^2 approximately. By increasing the number of nodes in environment, it falls to less than 200 μm^2.

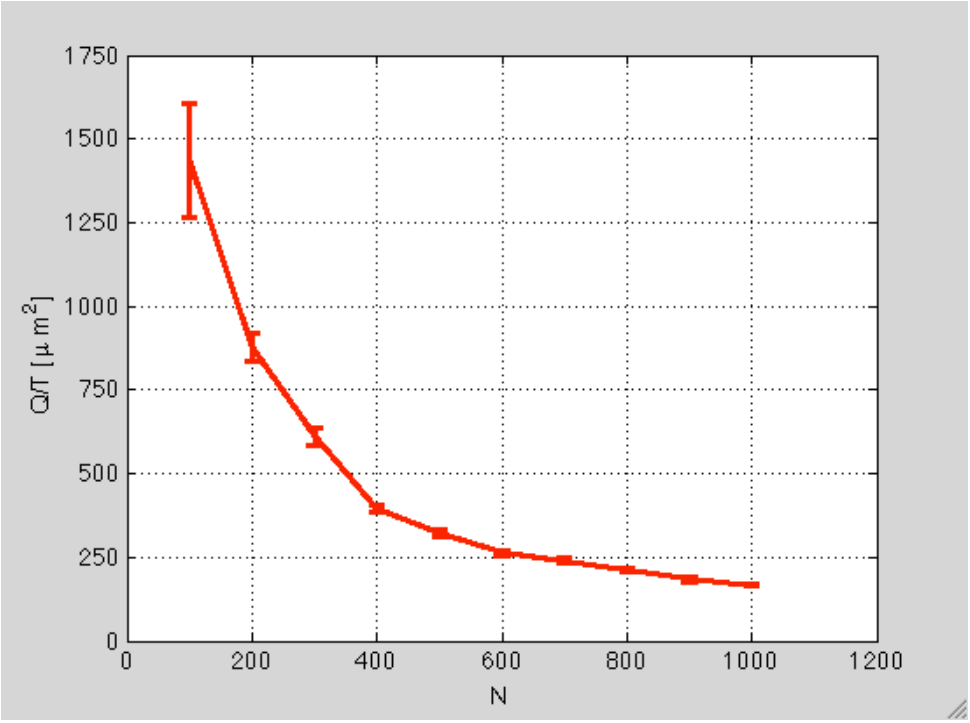


Figure 15: *Q/T* versus the number of nodes required to achieve 95% connectivity in 2D environment

The non-linear behavior in the three-dimensional environment is similar. As it is shown in Fig. 16, *Q/T* required decreases non-linearly from about 2,250,000 μm^3 with 100 nodes to approximately 30,000 μm^3 with 1000 nodes. In 3D, however, *Q/T*

required to achieve 95% connectivity with a given number of nodes and with fixed *T*, is larger.

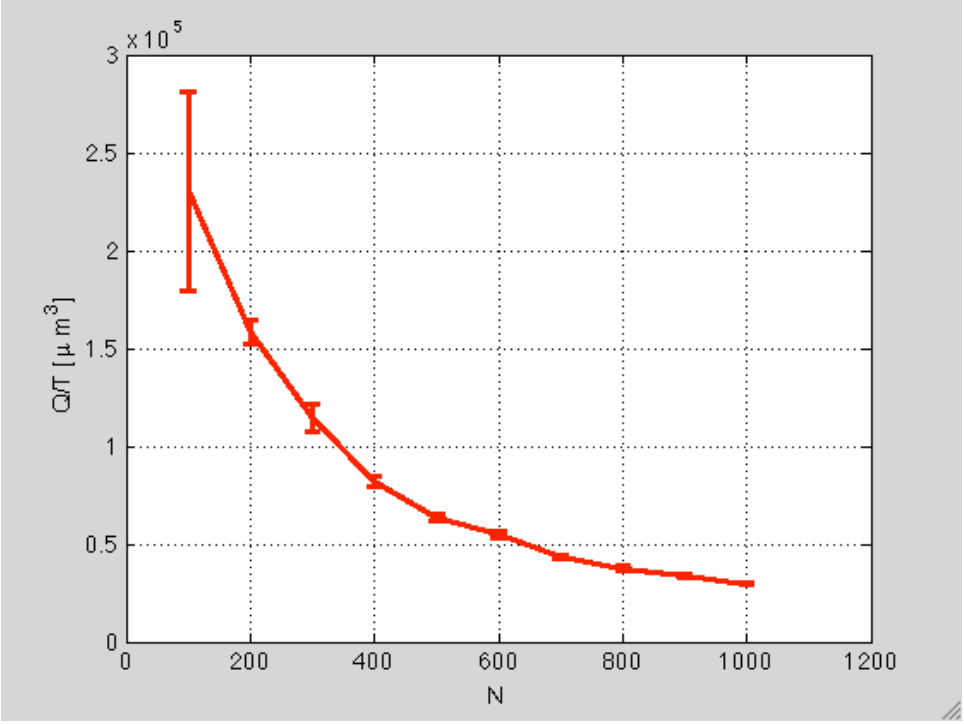


Figure 16: *Q/T* versus the number of nodes required to achieve 95% connectivity in 3D environment

4.4 *T/Q* Versus The Number Of Nodes Required To Achieve 95% Connectivity

The ratio of the proportion of detection threshold to the number of particles for having 95% connectivity has been evaluated again for different number of nodes ranging from 100 to 1000. As it was expected by increasing number of nodes, the *T/Q* ratio required also increases. This is because when there are more nodes in the area, the chance of hitting receiver is higher for particles so receivers even by greater detection thresholds can provide the connection between themselves and senders.

Though it is true that with smaller threshold values they can also have connectivity, the purpose here was to find the greatest possible detection threshold because the higher detection threshold results in reducing the effect of noise. The ratios are calculated based on 10 simulation runs. The standard error bars are also shown in the figures. As it can be seen from the Fig. 17, the behavior, unlike the previous case is almost linear, growing from 0.0005 μm^2 with 100 nodes to 0.006 μm^2 with 1000 nodes.

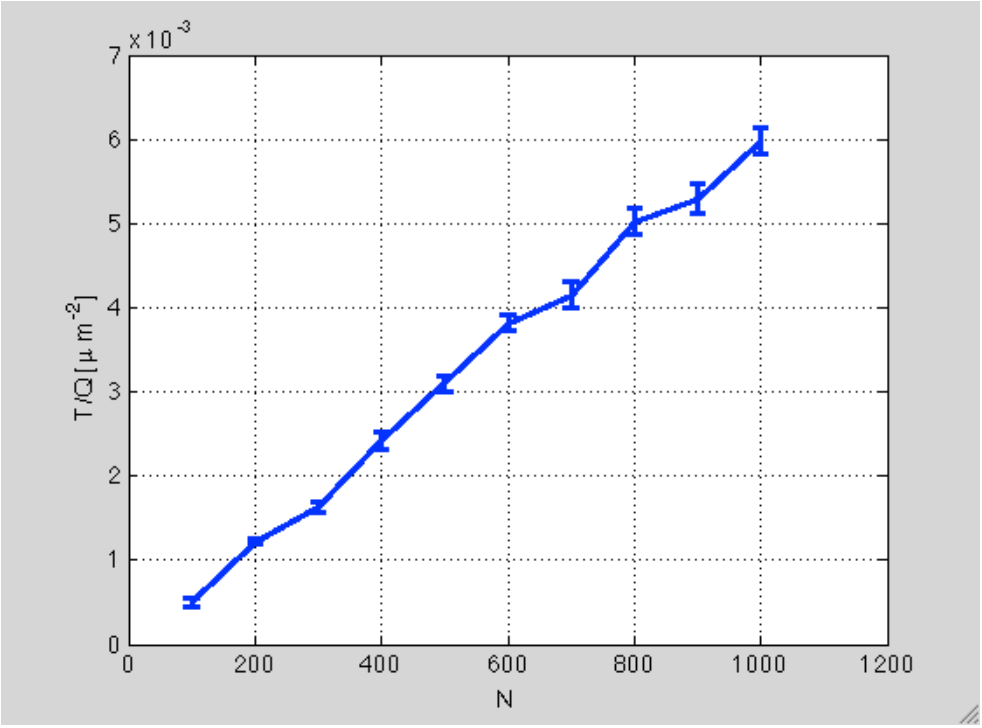


Figure 17: *T/Q* versus the number of nodes required to achieve 95% connectivity in 2D environment

In the three-dimensional environment, the overall results are similar as show in Fig. 18. However, the ratio *T/Q* for achieving 95% connectivity with a given number of nodes and a fixed *Q* must be smaller.

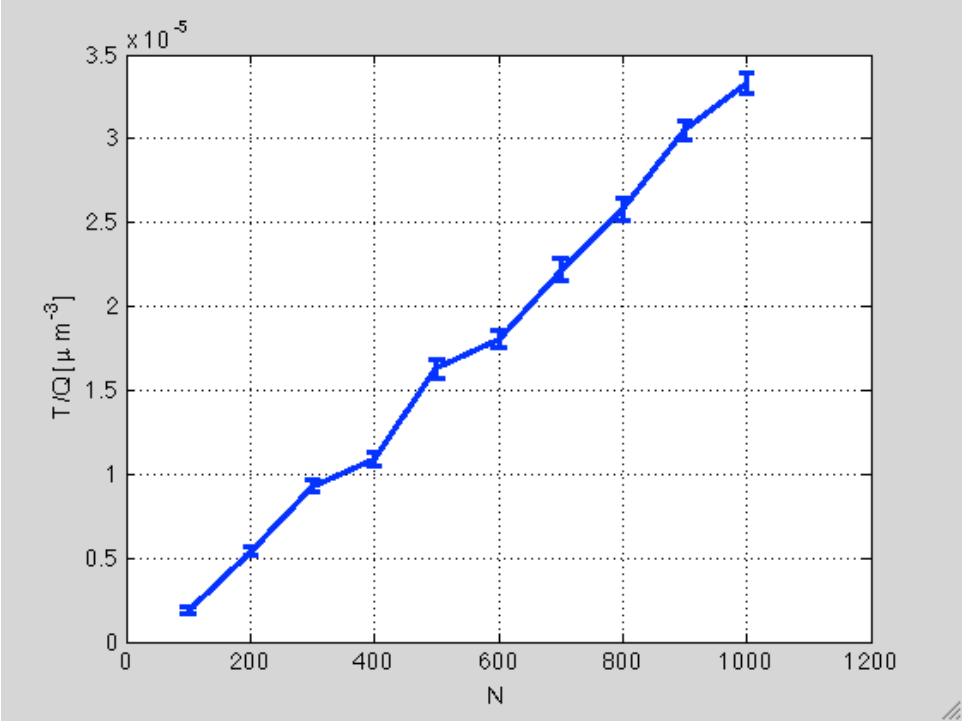


Figure 18: *T/Q* versus the number of nodes required to achieve 95% connectivity in 3D environment

4.5 *Q/T* Versus The Number Of Nodes Required To Achieve 100% Connectivity

The ratio *Q/T* required for having a fully connected network is also evaluated. As it can be easily observed from the Fig. 19, the ratio is higher compared with 95% connectivity. For 5% improvement in connectivity, the required number should approximately be doubled. These results and their corresponding error bars are also achieved with 10 simulation runs.

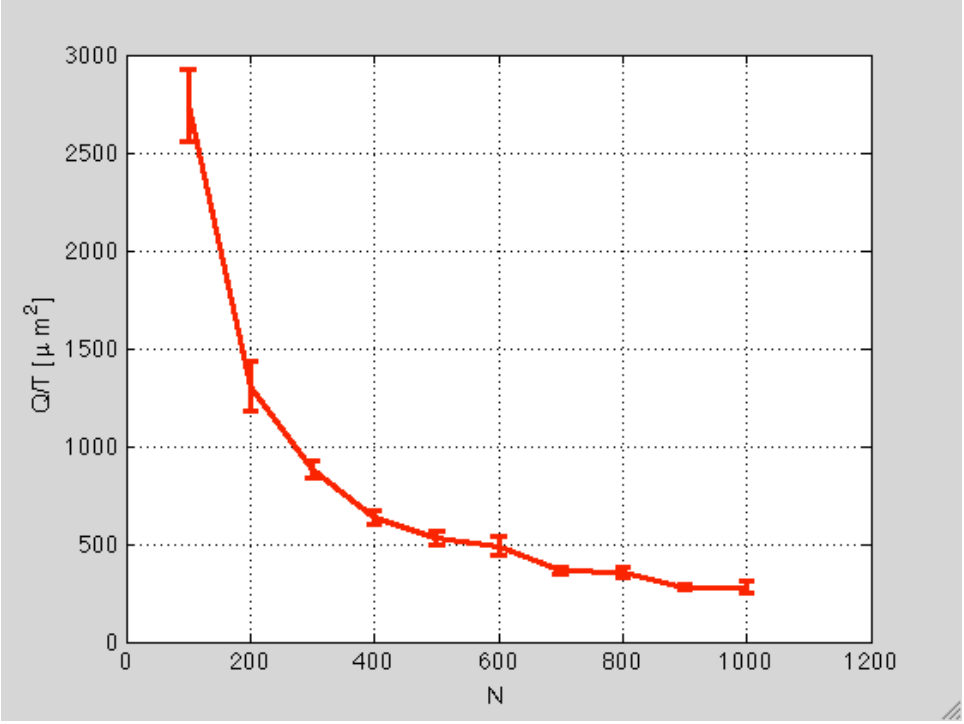


Figure 19: *Q/T* versus the number of nodes required to achieve 100% connectivity in 2D environment

Fig. 20 shows the same evaluation for three-dimensional environment. In 3D also for achieving 5% improvement in connectivity the required *Q/T* value should approximately be multiplied by a factor of 2.5.

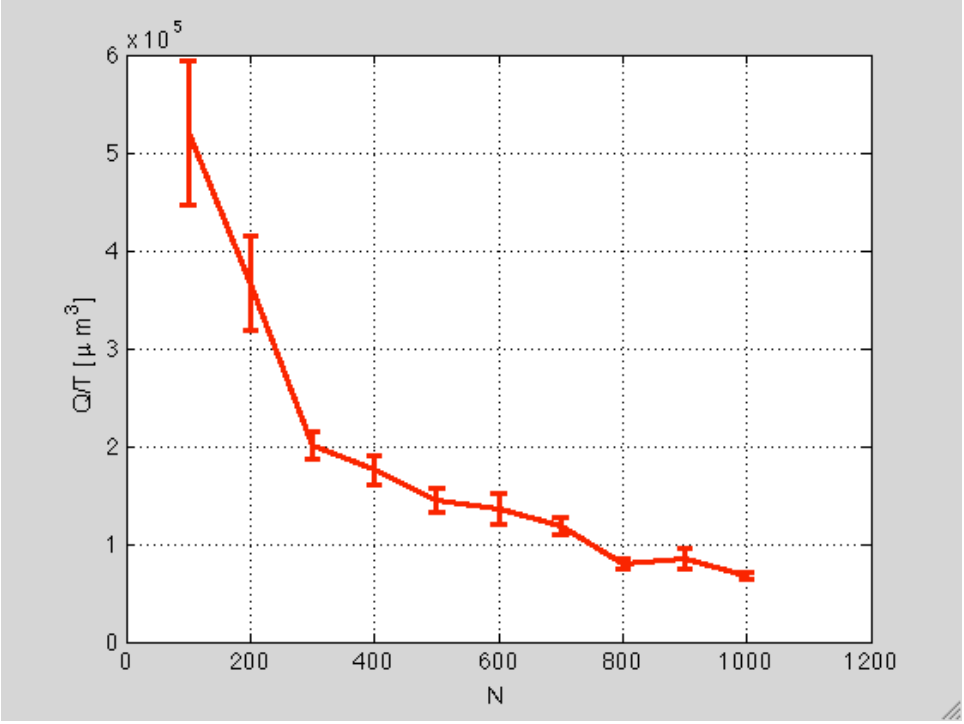


Figure 20: *Q/T* versus the number of nodes required to achieve 100% connectivity in 3D environment

4.6 *T/Q* Versus The Number Of Nodes Required To Achieve 100% Connectivity

The ratio *T/Q* needed for having a fully connected network is also analyzed in Fig. 21. Slight deviations from linearity may be due to insufficient number of runs for a 100% connected network, which is a strict condition in a finite region. However, the qualitative behavior of the graph was consistent with the expectations.

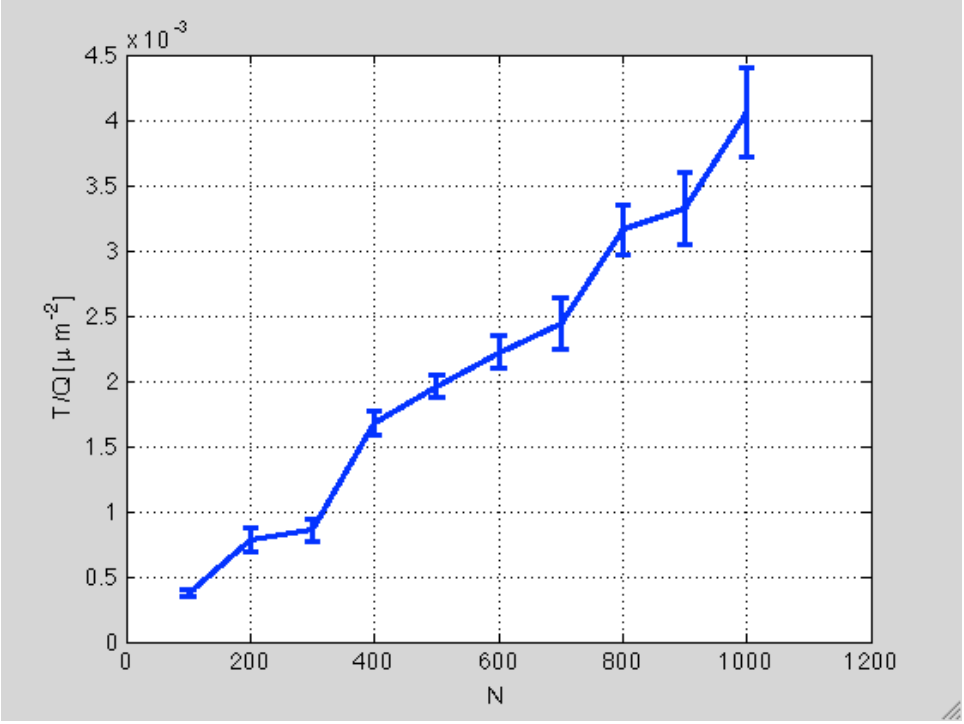


Figure 21: *T/Q* versus the number of nodes required to achieve 100% connectivity in 2D environment

The last figure, Fig. 22, belongs to the simulation for 3D environments.

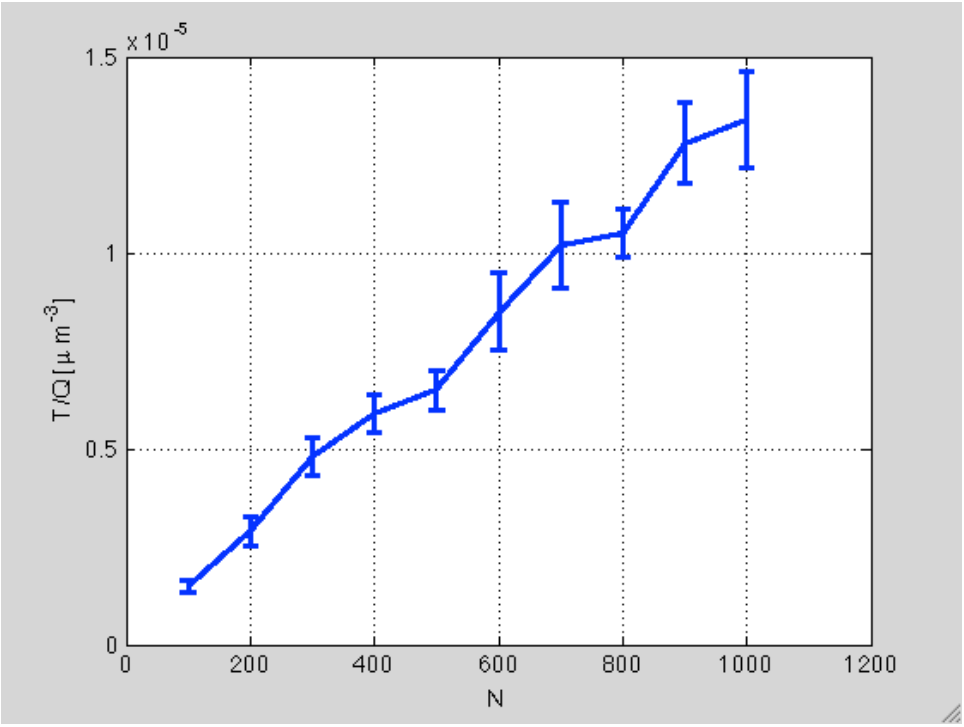


Figure 22: *T/Q* versus the number of nodes required to achieve 100% connectivity in 3D environment

Chapter 5

CONCLUSION

In this thesis, first, molecular communication as a biocompatible alternative to electromagnetic communication in nanonetworks has been introduced and its architecture and components have been explained. Then, results to assess the macroscopic behavior of nanonetworks, namely connectivity, have been presented for various scenarios. By generating random nodes in the environment by simulation, the percent connectivity as a function of various *Q/T* (number of released particles over detection threshold) and *T/Q* (detection threshold over number of released particles) values were evaluated. The results show that either by increasing *Q/T* or decreasing *T/Q* ratios, higher percent connectivity can be achieved. Then, with different number of nodes in the network, the required value of above ratios for having 95% and 100% connectivity were evaluated. The results illustrate that with higher number of nodes in the network, less *Q/T* is required for having 95% or 100% connectivity. In other words, by a fixed detection threshold value, fewer number of released particles is needed with higher number of nodes. Conversely, with higher *T/Q*, 95% or 100% connectivity can be achieved if the number of nodes in the network increases. In other words, with higher number of nodes in the network and fixed number of released particles, connected network can be achieved with a larger detection threshold value. In addition, for 5% improvement from 95% to 100% connectivity, the required ratios for *Q/T* and *T/Q* should be approximately doubled and halved respectively, which shows that 100% connectivity is a strict condition.

The main contribution of the thesis is to study diffusion-based molecular communication nanonetwork from macroscopic point of view, connectivity, and different factors affecting it. This study is expected to provide guidelines when designing nano-sensor networks in which connectivity is crucial.

Future work will involve further identification of differences between connectivity properties of traditional wireless ad hoc networks and nanonetworks as well as applying methods, such as using relay nodes, to improve the connectivity behavior of nanonetworks.

REFERENCES

[1] E. Drexler, 29 12 2009. [Online]. Available: http://www.metamodern.com.

[2] I. F. Akyildiz, F. Brunetti and C. Blázquez, "Nanonetworks: A new communication paradigm," *Computer Networks,* vol. 52, no. 12, p. 2260–2279, 2008.

[3] G. Whitesides, "The once and future nanomachine," *Scientific American,* pp. 78-83, 2001.

[4] E. Drexler, C. Peterson and G. Pergami, Unbounding the future: The nanotechnology revolution, Quill, 1991.

[5] T. Nakano, M. Moore, F. Wei, A. Vasilakos and J. Shuai, "Molecular Communication and Networking: Opportunities and Challenges," *IEEE Transactions on Nanobioscience,* vol. 11, no. 2, 2012.

[6] T. Nakano, M. Moore, A. Enomoto and T. Suda, "Molecular Communication Technology as a Biological ICT," in *Biological Functions for Information and Communication Technologies*, Springer Berlin Heidelberg, 2011, pp. 49-86.

[7] R. Freitas, "Pharmacytes: An ideal vehicle for targeted drug delivery," *Nanoscience and Nanotechnology,* vol. 6, pp. 2769-2775, 2006.

[8] O. Farokhzad and R. Langer, "Impact of nanotechnology on drug delivery,"

ACS Nano, vol. 3, no. 1, p. 16–20, 2009.

[9] T. Donaldson, "24th century medicine," *Cryonics,* pp. 16-34, 1988.

[10] D. Tessier, I. Radu and M. Filteau, "Antimicrobial fabrics coated with nano-sized silver salt crystals," *NSTI Nanotech,* vol. 1, pp. 762-764, 2005.

[11] R. Smalley, M. Dresselhaus, G. Dresselhaus and P. Avouris, Carbon Nanotubes: Synthesis, Structure, Properties and Applications, vol. 80, Springer, 2001.

[12] J. Han, J. Fu and R. Schoch, "Molecular sieving using nanofilters: past, present and future," *Lab on a Chip ,* vol. 8, no. 1, pp. 23-33, 2008.

[13] J. Oyekan, H. Hu and D. Gu, "Exploiting bacteria swarms for pollution mapping," in *Robotics and Biomimetics (ROBIO)*, 2009.

[14] I. Akyildiz and J. Jornet, "Electromagnetic wireless nanosensor networks," *Nano Communication Networks,* vol. 1, no. 1, pp. 3-19, 2010.

[15] S. Hiyama, Y. Moritani, T. Suda, R. Egashira, R. Enomoto, M. Moore and T. Nakano, "Molecular Communication," in *In Proc. of the 2005 NSTI Nanotechnology Conference*, 2005.

[16] L. Gine and I. Akyildiz, "Molecular communication options for long range nanonetworks," *Computer Networks,* vol. 53, no. 16, pp. 2753-2766, 2009.

[17] M. Moore, A. Enomoto, T. Suda, T. Nakano and Y. Okaie, "Molecular communication: New paradigm for communication among nano-scale biological machines," in *Handbook of Computer Networks: Distributed Networks, Network*

Planning, Control, Management, and New Trends and Applications, vol. 3, 2007, p. 1034–1054.

[18] S. Hiyama and Y. Moritani, "Molecular communication: Harnessing biochemical materials to engineer biomimetic communication sytems," *Nano Communication Networks,* vol. 1, no. 1, pp. 20-30, 2010.

[19] J. Smith, "The concept of information in biology," *Philosophy of Science,* vol. 67, no. 2, p. 177–194, 2000.

[20] J. Howard, Mechanics of motor proteins and the cytoskeleton, Sinauer Associates, 2001.

[21] G. Yang, Body sensor networks, Springer London, 2006.

[22] M. Kuran, H. Yilmaz, T. Tugcu and B. Ozerman, "Energy model for communication via diffusion in nanonetworks," *Nano Communication Networks,* vol. 1, no. 2, pp. 86-95, 2010.

[23] M. Moore, A. Enomoto and T. Nakano, "A design of a molecular communication system for nanomachines using molecular motors," in *Pervasive Computing and Communications Workshops*, 2006.

[24] M. Gregori and I. Akyildiz, "A New NanoNetwork Architecture Using Flagellated Bacteria and Catalytic Nanomotors," *IEEE Journal On Selected Areas In Communications,* vol. 28, no. 4, pp. 612-619, 2010.

[25] L. Giné and I. Akyildiz, "Molecular Communication Options for Long Range

Nanonetworks," *Computer Networks: The International Journal of Computer and Telecommunications Networking,* vol. 53, no. 16, pp. 2753-2766, 2009.

[26] T. Nakano, T. Suda and M. Moore, “Molecular communication for nanomachines using intercellular calcium signaling,” in *Fifth IEEE Conference on Nanotechnology*, 2005.

[27] T. Kievit and B. Iglewski, "Bacterial quorum sensing in pathogenic relationships," *Infection and Immunity,* vol. 68, no. 9, p. 4839–4849, 2000.

[28] K. Oiwa and H. Sakakibara, "Recent progress in dynein structure and mechanism," *Current Opinion in Cell Biology,* vol. 17, no. 1, pp. 98-103, 2005.

[29] T. Shima, T. Kon, K. Imamula, R. Ohkura and K. Sutoh, "Two modes of microtubule sliding driven by cytoplasmic dynein," *Proceedings of the National Academy of Sciences,* vol. 103, no. 47, pp. 7736-17740, 2006.

[30] S. Toba and K. Oiwa, "Swing or embrace? New aspects of motility inspired by dynein structure in situ.," *Bioforum Europe,* vol. 10, p. 14–16, 2006.

[31] J. Berg, J. Tymoczko and L. Stryer, Biochemisty, New York: W. H. Freeman, 2002.

[32] R. Tamarin, Principles of genetics, New York: McGraw-Hill, 1999.

[33] L. Cobo and I. Akyildiz, "Bacteria-based communication in nanonetworks," *Nano Communication Networks,* vol. 1, no. 4, pp. 244-256, 2010.

[34] T. Suda, M. Moore, T. Nakano, R. Egashira and A. Enomoto, "Exploratory research on molecular communication between nanomachines," in *Genetic and Evolutionary Computation Conference (GECCO)*, 2005.

[35] Y. Moritani, S. Hiyama and T. Suda, "Molecular communication among nanomachines using vesicles," in *NSTI Nanotechnology Conference and Trade Show*, 2006.

[36] N. R. Lacasa, "Modeling the Molecular Communication Nanonetworks," MS Thesis, Universitat Politècnica de Catalunya, 2009.

[37] M. S. Kuran, H. B. Yilmaz, T. Tugcu and I. F. Akyildiz, "Modulation techniques for communication via diffusion in nanonetworks," *IEEE International Conference on Communications (ICC),* pp. 1-5, 2011.

[38] T. Schneider, "Theory of molecular machines I. Channel capacity of molecular machines," *Theoretical Biology,* pp. 83-123, 1991.

[39] M. Pierobon and I. Akyildiz, "A physical end-to-end model for molecular communication in nanonetworks," *IEEE Journal on Selected Areas in Communications,* vol. 28, no. 4, pp. 602-611, 2010.

[40] B. Atakan and O. Akan, "Deterministic capacity of information flow in molecular nanonetworks," *Nano Communication Networks,* vol. 1, no. 1, pp. 31-42, 2010.

[41] J. Philibert, "One and a half century of diffusion: Fick, Einstein, before and

beyond," *Diffusion Fundamentals ,* vol. 2, 2005.

[42] M. Kuran, H. Yilmaz, T. Tugcu and I. Akyildiz, "Interference effects on modulation techniques in diffusion based nanonetworks," *Nano Communication Networks,* vol. 3, no. 1, pp. 65-73, 2012.

[43] T. Nakano and J. Shuai, "Repeater Design and Modeling for Molecular Communication Networks," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2011.

[44] A. Fereidouni and D. Arifler, "Analysis of connectivity in diffusion-based molecular nano communication networks," 4th NaNoNetworking Summit, Barcelona, Spain, June 2012. [Online]. Available: http://www.n3cat.upc.edu/.

[45] R. Diestel, Graph Theory, Springer, 2010.

[46] M. Haenggi, J. Andrews, F. Baccelli, O. Dousse and M. Franceschetti, "Stochastic geometry and random graphs for the analysis and design of wireless networks," *IEEE J. Select. Areas Commun.,* vol. 27, no. 7, p. 1029–1046, 2009.

[47] D. Stoyan, W. Kendall and J. Mecke, Stochastic Geometry and Its Applications, Wiley; 2 edition, 1995.

[48] R. Freitas, Nanomedicine, Volume I: Basic Capabilities, Landes Biosience, 1999.

[49] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts and P. Walter, Molecular biology of the cell, Garland Science, 2002.

APPENDIX

Appendix A: Source Codes For Simulations

Connectivity_vs number of released particles in 2D

```
function [par_arr,prc_arr] = calc_conn_par( n,trs,par )

%This function calculates the connectivity percentage for
%different number of emitted particles in 2D environment

par_arr=[] ;
prc_arr=[] ;

x=unifrnd(0,100,n,1) ;   %generates random points in 100x100 area
y=unifrnd(0,100,n,1) ;
Rnd=[x y] ;
[d]=pdist(Rnd) ;         %measures the distance between nodes
sq=squareform(d) ;


adj=zeros(n) ;
p=par ;


    while par<10*p   %this loop increases number of emitted
particles
                     % and finds the corresponding connectivity
          k=par/(pi*exp(1)) ;

          for ii=1:n       %constructs adjacent matrix for specific
              for jj=1:n   %number of emitted particles

                  if (k/(sq(ii,jj).^2)>=trs && ii~=jj);
                      adj(ii,jj)=1 ;
                  else
                      adj(ii,jj)=0 ;

                  end
              end
          end

          spa=sparse(adj);

          [NUM,COM]=graphconncomp(spa) ;


          unv = unique(COM);       % finds max no of connected nodes
          rep = histc(COM,unv);
          Mx= max(rep) ;
          prc=(Mx/n)*100  ;      %finds connectivity percentage

          par_arr=[par_arr;par] ;
          prc_arr=[prc_arr;prc] ;

    par=par+(p/10) ;   %increases number of particles by 10%
```

```
    end

end
```

```
% Percent connectivity as a function of number of released particles
% Different curves for different number of nodes

clear all
set(0, 'defaultaxesfontsize', 14);
set(0, 'defaultlinelinewidth', 2);

Q=10000;
T=100;
[x100,y100]=calc_conn_par(100,T,Q) ;
[x200,y200]=calc_conn_par(200,T,Q) ;
[x300,y300]=calc_conn_par(300,T,Q) ;
[x400,y400]=calc_conn_par(400,T,Q) ;
[x500,y500]=calc_conn_par(500,T,Q) ;

plot(x100/T,y100,'-r',x200/T,y200,'--
g',x300/T,y300,':b',x400/T,y400,'-.k',x500/T,y500,'.-m')
legend('N=100','N=200','N=300','N=400','N=500','Location',
'Northwest')
xlabel('Q/T [\mum^{2}]') ;
ylabel('% Connectivity') ;
set(gca,'XTick',0:100:1000);
set(gca,'YTick',0:10:100);
grid on ;
```

Connectivity_vs number of released particles in 3D

```
function [par_arr,prc_arr] = calc_conn_par_3d( n,trs,par )

%This function calculates the connectivity percentage for
%different number of emitted particles
par_arr=[] ;
prc_arr=[] ;

x=unifrnd(0,100,n,1) ;   %generates random points in 100x100 area
y=unifrnd(0,100,n,1) ;
z=unifrnd(0,100,n,1) ;
Rnd=[x y z] ;
[d]=pdist(Rnd) ;         %measures the distance between nodes
sq=squareform(d) ;
adj=zeros(n) ;
p=par ;


    while par<10*p   %this loop increases number of emitted
```

```
                    % and finds the corresponding connectivity

            k=(3*par)/(2*pi*exp(1)).^1.5 ;

            for ii=1:n       %constructs adjacent matrix for specific
                for jj=1:n   %number of emitted particles

                    if (k/(sq(ii,jj).^3)>=trs && ii~=jj);
                        adj(ii,jj)=1 ;
                    else
                        adj(ii,jj)=0 ;

                    end
                end
            end

            spa=sparse(adj);

            [NUM,COM]=graphconncomp(spa) ;


            unv = unique(COM);      % finds max no of connected nodes
            rep = histc(COM,unv);
            Mx= max(rep) ;
            prc=(Mx/n)*100  ;      %finds connectivity percentage

            par_arr=[par_arr;par] ;
            prc_arr=[prc_arr;prc] ;




        par=par+(p/10) ;   %increases number of particles by 10%

        end

    end



% Percent connectivity as a function of number of released particles
% Different curves for different number of nodes

clc
clear all
set(0, 'defaultaxesfontsize', 14);
set(0, 'defaultlinelinewidth', 2);

Q=1000000;
T=100;

[x100,y100]=calc_conn_par_3d(100,T,Q) ;
[x200,y200]=calc_conn_par_3d(200,T,Q) ;
[x300,y300]=calc_conn_par_3d(300,T,Q) ;
[x400,y400]=calc_conn_par_3d(400,T,Q) ;
[x500,y500]=calc_conn_par_3d(500,T,Q) ;

figure
plot(x100/T,y100,'-r',x200/T,y200,'--
```

```
g',x300/T,y300,':b',x400/T,y400,'-.k',x500/T,y500,'.-m')
hold on ;

legend('N=100','N=200','N=300','N=400','N=500','Location',
'Northwest')

xlabel('Q/T [\mum^{3}]') ;
ylabel('% Connectivity') ;
set(gca,'YTick',0:10:100);
grid on ;
```

Connectivity_vs detection threshold in 2D

```
function [ trs_arr,prc_arr ] = calc_conn_trs( n,trs,par )
%This function calculates the connectivity percentage for different
%concentration thresholds

trs_arr=[] ;
prc_arr=[] ;

x=unifrnd(0,100,n,1) ;   %generates random points in 100x100 area
y=unifrnd(0,100,n,1) ;
Rnd=[x y] ;
[d]=pdist(Rnd) ;         %measures the distance between nodes
sq=squareform(d) ;

adj=zeros(n) ;
k=par/(pi*exp(1)) ;


  while trs>0      %this loop decreases the concentration threshold
                   %and finds the corresponding connectivity
    for ii=1:n
                       %constructs the adjacent matrix for specific
        for jj=1:n     %concentration threshold

            if (k/(sq(ii,jj).^2)>=trs && ii~=jj);
               adj(ii,jj)=1 ;
            else
               adj(ii,jj)=0 ;
            end
        end

    end

    spa=sparse(adj);

    [NUM,COM]=graphconncomp(spa) ;


    unv = unique(COM);
    rep = histc(COM,unv);
    Mx= max(rep) ;         %finds max number of connected nodes
```

```
    prc=(Mx/n)*100 ;        %finds the connectivity percentage


    trs_arr=[trs_arr;trs] ;
    prc_arr=[prc_arr;prc] ;



    trs=trs-1 ;      %decreases 1 unit from concentration threshold


  end


end



% Percent connectivity as a function of threshold
% Different curves for different number of nodes

clear all
set(0, 'defaultaxesfontsize', 14);
set(0, 'defaultlinelinewidth', 2);

Q=10000;
T=100;
[x100,y100]=calc_conn_trs(100,T,Q) ;
[x200,y200]=calc_conn_trs(200,T,Q) ;
[x300,y300]=calc_conn_trs(300,T,Q) ;
[x400,y400]=calc_conn_trs(400,T,Q) ;
[x500,y500]=calc_conn_trs(500,T,Q) ;

plot(x100/Q*1000,y100,'-r',x200/Q*1000,y200,'--
g',x300/Q*1000,y300,':b',x400/Q*1000,y400,'-.k',x500/Q*1000,y500,'.-
m')
legend('N=100','N=200','N=300','N=400','N=500')
xlabel('T/Q (x10^3) [\mum^{-2}]') ;
ylabel('% Connectivity') ;
set(gca,'XTick',0:1:10);
set(gca,'YTick',0:10:100);
grid on ;
```

Connectivity_vs detection threshold in 3D

```
function [ trs_arr,prc_arr ] = calc_conn_trs_3d( n,trs,par )
%This function calculates the connectivity percentage for different
%concentration thresholds

trs_arr=[] ;
prc_arr=[] ;

x=unifrnd(0,100,n,1) ;   %generates random points in 100x100 area
y=unifrnd(0,100,n,1) ;
z=unifrnd(0,100,n,1) ;
```

```
Rnd=[x y z] ;
[d]=pdist(Rnd) ;          %measures the distance between nodes
sq=squareform(d) ;

adj=zeros(n) ;
k=(3*par)/(2*pi*exp(1)).^1.5 ;

  while trs>0       %this loop decreases the concentration threshold
                    %and finds the corresponding connectivity
    for ii=1:n
                        %constructs the adjacent matrix for specific
        for jj=1:n      %concentration threshold

            if (k/(sq(ii,jj).^3)>=trs && ii~=jj);
               adj(ii,jj)=1 ;
            else
               adj(ii,jj)=0 ;
            end
        end

    end

    spa=sparse(adj);

    [NUM,COM]=graphconncomp(spa) ;


    unv = unique(COM);
    rep = histc(COM,unv);
    Mx= max(rep) ;          %finds max number of connected nodes
    prc=(Mx/n)*100 ;        %finds the connectivity percentage


    trs_arr=[trs_arr;trs] ;
    prc_arr=[prc_arr;prc] ;



    trs=trs-1 ;       %decreases 1 unit from concentration threshold


  end


end



% Percent connectivity as a function of threshold
% Different curves for different number of nodes

clc
clear all
set(0, 'defaultaxesfontsize', 14);
set(0, 'defaultlinelinewidth', 2);

Q=1000000;
T=100;
```

```
[x100,y100]=calc_conn_trs_3d(100,T,Q) ;
[x200,y200]=calc_conn_trs_3d(200,T,Q) ;
[x300,y300]=calc_conn_trs_3d(300,T,Q) ;
[x400,y400]=calc_conn_trs_3d(400,T,Q) ;
[x500,y500]=calc_conn_trs_3d(500,T,Q) ;

figure
plot(x100/Q*1000,y100,'-r',x200/Q*1000,y200,'--
g',x300/Q*1000,y300,':b',x400/Q*1000,y400,'-.k',x500/Q*1000,y500,'.-
m')
hold on ;

legend('N=100','N=200','N=300','N=400','N=500')

xlabel('T/Q [\mum^{-3}]') ;
ylabel('% Connectivity') ;
set(gca,'YTick',0:10:100);
grid on ;
```

Number of nodes vs number of emitted particles for 95% connectivity in 2D

```
function [par_arr,nod_arr,err_arr] =
calc_node_par(first,last,rpt,trs)
%This function calculates number of nodes vs number of emitted
%particles in which there is more than 95% of connectivity
%with a given concentration threshold and number of repetition

par_arr=[] ;
nod_arr=[] ;
err_arr=[] ;

incr=100 ;

total=((last-first)/incr)+1 ;

R=zeros(rpt,total) ;
count=0 ;


for n=first:incr:last

      count=count+1 ;

      for ll=1:rpt

    x=unifrnd(0,100,n,1) ;   %generates random points in 100x100
area
    y=unifrnd(0,100,n,1) ;
    Rnd=[x y] ;
    [d]=pdist(Rnd) ;         %measures the distance between nodes
    sq=squareform(d) ;

    adj=zeros(n);
    par=10000 ;              %sets no. of emitted particles to 10000
```

```
    b=par ;

    while par<20*b

          k=par/(pi*exp(1)) ;



          for ii=1:n          %constructs the adjacent matrix
              for jj=1:n
                  if (k/(sq(ii,jj).^2)>=trs && ii~=jj);
                     adj(ii,jj)=1 ;
                  else
                     adj(ii,jj)=0 ;
                  end
              end
          end

          spa=sparse(adj);

          [NUM,COM]=graphconncomp(spa) ; %finds the connected
          unv = unique(COM);             %components of graph
          rep = histc(COM,unv);
          Mx= max(rep) ;
          prc=(Mx/n)*100 ;          %finds the connectivity
percentage

          if prc>95
             R(ll,count)=par ;

          break

          else

              par=par+(b/10) ;   %increases number of particles by
10%

          end


    end

      end



end

M=mean(R) ;
E=std(R)/sqrt(rpt) ;


cc=0 ;

  for  ii=first:incr:last
       cc=cc+1 ;
```

```
        par_arr=[par_arr;ii] ;
        nod_arr=[nod_arr;M(cc)] ;
        err_arr=[err_arr;E(cc)] ;

  end



end



% Determines the number of particles necessary for achieving 95%
connectivity
% Normalized number of particles is plotted vs. number of nodes

clear all
set(0, 'defaultaxesfontsize', 14);
set(0, 'defaultlinelinewidth', 2);
T=100;
ITER=10;
[x,y,e]=calc_node_par(100,1000,ITER,T) ;

errorbar(x,y/T,e/T,'-r')
xlabel('N') ;
ylabel('Q/T [\mum^{2}]') ;
set(gca,'YTick',0:250:1750)
grid on ;
```

Number of nodes vs number of emitted particles for 95% connectivity in 3D

```
function [par_arr,nod_arr,err_arr] =
calc_node_par_3d(first,last,rpt,trs)
%This function calculates number of nodes vs number of emitted
%particles in which there is more than 95% of connectivity
%with a given concentration threshold and number of repetition

par_arr=[] ;
nod_arr=[] ;
err_arr=[] ;

incr=100 ;

total=((last-first)/incr)+1 ;

R=zeros(rpt,total) ;
count=0 ;


for n=first:incr:last

      count=count+1 ;

      for ll=1:rpt
```

```
    x=unifrnd(0,100,n,1) ;   %generates random points in 100x100
area
    y=unifrnd(0,100,n,1) ;
    z=unifrnd(0,100,n,1) ;

    Rnd=[x y z] ;
    [d]=pdist(Rnd) ;         %measures the distance between nodes
    sq=squareform(d) ;

    adj=zeros(n);
    par=1000000 ;              %sets no. of emitted particles to
10000
    b=par ;

    while par<40*b

          k=(3*par)/(2*pi*exp(1)).^1.5 ;



          for ii=1:n         %constructs the adjacent matrix
              for jj=1:n
                  if (k/(sq(ii,jj).^3)>=trs && ii~=jj);
                     adj(ii,jj)=1 ;
                  else
                     adj(ii,jj)=0 ;
                  end
              end
          end

          spa=sparse(adj);

          [NUM,COM]=graphconncomp(spa) ; %finds the connected
          unv = unique(COM);             %components of graph
          rep = histc(COM,unv);
          Mx= max(rep) ;
          prc=(Mx/n)*100 ;        %finds the connectivity percentage

          if prc>95
             R(ll,count)=par ;

          break

          else

              par=par+(b/10) ; %increases number of particles by 10%

          end


    end

      end
```

```
end

M=mean(R) ;
E=std(R)/sqrt(rpt) ;


cc=0 ;

  for  ii=first:incr:last
       cc=cc+1 ;

       par_arr=[par_arr;ii] ;
       nod_arr=[nod_arr;M(cc)] ;
       err_arr=[err_arr;E(cc)] ;

  end



end



% Determines the number of particles necessary for achieving 95%
connectivity
% Normalized number of particles is plotted vs. number of nodes


clc
clear all

set(0, 'defaultaxesfontsize', 14);
set(0, 'defaultlinelinewidth', 2);

T=100;
ITER=10;
[x,y,e]=calc_node_par_3d(100,1000,ITER,T) ;

figure
errorbar(x,y/T,e/T,'-r')
hold on ;


xlabel('N') ;
ylabel('Q/T [\mum^{3}]') ;
grid on ;
```

Number of nodes vs number of emitted particles for 100% connectivity in 3D

```
function [par_arr,nod_arr,err_arr] =
calc_node_par_3d_100(first,last,rpt,trs)
%This function calculates number of nodes vs number of emitted
%particles in which there is more than 100% of connectivity
%with a given concentration threshold and number of repetition

par_arr=[] ;
```

```
nod_arr=[] ;
err_arr=[] ;

incr=100 ;

total=((last-first)/incr)+1 ;

R=zeros(rpt,total) ;
count=0 ;


for n=first:incr:last

      count=count+1 ;

      for ll=1:rpt

    x=unifrnd(0,100,n,1) ;   %generates random points in 100x100
area
    y=unifrnd(0,100,n,1) ;
    z=unifrnd(0,100,n,1) ;

    Rnd=[x y z] ;
    [d]=pdist(Rnd) ;         %measures the distance between nodes
    sq=squareform(d) ;

    adj=zeros(n);
    par=2000000 ;          %sets no. of emitted particles to 2000000
    b=par ;

    while par<50*b

          k=(3*par)/(2*pi*exp(1)).^1.5 ;



          for ii=1:n         %constructs the adjacent matrix
              for jj=1:n
                  if (k/(sq(ii,jj).^3)>=trs && ii~=jj);
                     adj(ii,jj)=1 ;
                  else
                     adj(ii,jj)=0 ;
                  end
              end
          end

          spa=sparse(adj);

          [NUM,COM]=graphconncomp(spa) ; %finds the connected
          unv = unique(COM);             %components of graph
          rep = histc(COM,unv);
          Mx= max(rep) ;
          prc=(Mx/n)*100 ;       %finds the connectivity percentage

          if prc==100
             R(ll,count)=par ;

          break
```

```
            else

                par=par+(b/10) ; %increases number of particles by 10%

            end


        end

          end



end

M=mean(R) ;
E=std(R)/sqrt(rpt) ;


cc=0 ;

  for  ii=first:incr:last
       cc=cc+1 ;

       par_arr=[par_arr;ii] ;
       nod_arr=[nod_arr;M(cc)] ;
       err_arr=[err_arr;E(cc)] ;

  end



end



% Determines the number of particles necessary for achieving 100%
connectivity
% Normalized number of particles is plotted vs. number of nodes

clc
clear all

set(0, 'defaultaxesfontsize', 14);
set(0, 'defaultlinelinewidth', 2);

T=100;
ITER=10;
[x,y,e]=calc_node_par_3d_100(100,1000,ITER,T) ;

figure
errorbar(x,y/T,e/T,'-r')
hold on ;

xlabel('N') ;
ylabel('Q/T [\mum^{3}]') ;
grid on ;
```

Number of nodes vs number of emitted particles for 100% connectivity in 2D

```
function [par_arr,nod_arr,err_arr] =
calc_node_par_100(first,last,rpt,trs)
%This function calculates number of nodes vs number of emitted
%particles in which there is more than 100% of connectivity
%with a given concentration threshold and number of repetition

par_arr=[] ;
nod_arr=[] ;
err_arr=[] ;

incr=100 ;

total=((last-first)/incr)+1 ;

R=zeros(rpt,total) ;
count=0 ;


for n=first:incr:last

      count=count+1 ;

      for ll=1:rpt

    x=unifrnd(0,100,n,1) ;   %generates random points in 100x100
area
    y=unifrnd(0,100,n,1) ;
    Rnd=[x y] ;
    [d]=pdist(Rnd) ;         %measures the distance between nodes
    sq=squareform(d) ;

    adj=zeros(n);
    par=10000 ;              %sets no. of emitted particles to 10000
    b=par ;

    while par<40*b

          k=par/(pi*exp(1)) ;
          for ii=1:n         %constructs the adjacent matrix
              for jj=1:n
                  if (k/(sq(ii,jj).^2)>=trs && ii~=jj);
                     adj(ii,jj)=1 ;
                  else
                     adj(ii,jj)=0 ;
                  end
              end
          end

          spa=sparse(adj);

          [NUM,COM]=graphconncomp(spa) ; %finds the connected
          unv = unique(COM);             %components of graph
          rep = histc(COM,unv);
          Mx= max(rep) ;
          prc=(Mx/n)*100 ;         %finds the connectivity
```

```
percentage

          if prc==100
             R(ll,count)=par ;

          break

          else

              par=par+(b/10) ;   %increases number of particles by
10%

          end


    end

      end



end

M=mean(R) ;
E=std(R)/sqrt(rpt) ;


cc=0 ;

  for  ii=first:incr:last
       cc=cc+1 ;

       par_arr=[par_arr;ii] ;
       nod_arr=[nod_arr;M(cc)] ;
       err_arr=[err_arr;E(cc)] ;

  end



end
% Determines the number of particles necessary for achieving 100%
connectivity
% Normalized number of particles is plotted vs. number of nodes

clc
clear all

set(0, 'defaultaxesfontsize', 14);
set(0, 'defaultlinelinewidth', 2);

T=100;
ITER=10;
[x,y,e]=calc_node_par_100(100,1000,ITER,T) ;

figure
errorbar(x,y/T,e/T,'-r')
hold on ;
```

```
xlabel('N') ;
ylabel('Q/T [\mum^{2}]') ;
grid on ;
```

Number of nodes vs detection threshold for 95% connectivity in 2D

```
function [ trs_arr,nod_arr,err_arr ] = calc_node_trs(
first,last,rpt,par)
%This function calculates concentration threshold vs number of nodes
%in which there is more than 95% connectivity
%with a given number of emitted particles and number of repetition


trs_arr=[] ;
nod_arr=[] ;
err_arr=[] ;

incr=100 ;
total=((last-first)/incr)+1 ;

R=zeros(rpt,total) ;
count=0 ;


k=par/(pi*exp(1)) ;


   for n=first:incr:last

          count=count+1 ;

          for ll=1:rpt

              x=unifrnd(0,100,n,1) ; %generates random points in
100x100 area
              y=unifrnd(0,100,n,1) ;
              Rnd=[x y] ;
              [d]=pdist(Rnd) ;  %measures the distance between nodes
              sq=squareform(d) ;


              trs=100 ;         %sets concentration threshold to 100
             adj=zeros(n) ;

              while trs>0
                    for ii=1:n      %constructs the adjacent matrix
                        for jj=1:n
                            if (k/(sq(ii,jj).^2)>=trs && ii~=jj);
                               adj(ii,jj)=1 ;
                            else
                               adj(ii,jj)=0 ;
                            end
                        end
                    end
```

```
                    spa=sparse(adj);

                    [NUM,COM]=graphconncomp(spa) ;
%finds the  connected components of graph
                    unv = unique(COM);
                    rep = histc(COM,unv);
                    Mx= max(rep) ;
                    prc=(Mx/n)*100 ;
%finds the connectivity percentage

                    if prc>95
                       R(ll,count)=trs ;

                       break

                    else

                      trs=trs-1 ;
%decreases 1 unit from concentration threshold

                    end


                end


            end


        end

M=mean(R) ;
E=std(R)/sqrt(rpt) ;


cc=0 ;

  for  ii=first:incr:last
     cc=cc+1 ;

     trs_arr=[trs_arr;ii] ;
     nod_arr=[nod_arr;M(cc)] ;
     err_arr=[err_arr;E(cc)] ;

  end

end




% Determines the threshold necessary for achieving 95% connectivity
% Normalized threshold is plotted vs. number of nodes
% Area = 100 micrometer x 100 micrometer
```

```
clear all
set(0, 'defaultaxesfontsize', 14);
set(0, 'defaultlinelinewidth', 2);
Q=10000;
ITER=10;
[x,y,e]=calc_node_trs(100,1000,ITER,Q) ;

errorbar(x,y/Q,e/Q,'-b')
xlabel('N') ;
ylabel('T/Q [\mum^{-2}]') ;
grid on ;
```

Number of nodes vs detection threshold for 95% connectivity in 3D

```
function [trs_arr,nod_arr,err_arr ] =
calc_node_trs_3d(first,last,rpt,par )
%This function calculates concentration threshold vs number of nodes
%in which there is more than 95% connectivity
%with a given number of emitted particles and number of repetition


trs_arr=[] ;
nod_arr=[] ;
err_arr=[] ;

incr=100 ;
total=((last-first)/incr)+1 ;

R=zeros(rpt,total) ;
count=0 ;


k=(3*par)/(2*pi*exp(1)).^1.5 ;


   for n=first:incr:last

          count=count+1 ;

          for ll=1:rpt

              x=unifrnd(0,100,n,1) ;  %generates random points in
100x100 area
              y=unifrnd(0,100,n,1) ;
              z=unifrnd(0,100,n,1) ;

              Rnd=[x y z] ;
              [d]=pdist(Rnd) ;    %measures the distance between
nodes
              sq=squareform(d) ;


              trs=100 ;         %sets concentration threshold to 100
```

```
            adj=zeros(n) ;

             while trs>0
                   for ii=1:n      %constructs the adjacent matrix
                       for jj=1:n
                           if (k/(sq(ii,jj).^3)>=trs && ii~=jj);
                              adj(ii,jj)=1 ;
                           else
                              adj(ii,jj)=0 ;
                           end
                       end
                    end

                    spa=sparse(adj);

                   [NUM,COM]=graphconncomp(spa) ;
                   unv = unique(COM);
                   rep = histc(COM,unv);
                   Mx= max(rep) ;
                   prc=(Mx/n)*100 ;
%finds the connectivity percentage

                   if prc>95
                      R(ll,count)=trs ;

                      break

                   else

                     trs=trs-1 ;
%decreases 1 unit from concentration threshold

                   end


              end


         end
end

M=mean(R) ;
E=std(R)/sqrt(rpt) ;


cc=0 ;

  for  ii=first:incr:last
     cc=cc+1 ;

     trs_arr=[trs_arr;ii] ;
     nod_arr=[nod_arr;M(cc)] ;
     err_arr=[err_arr;E(cc)] ;

  end

end
```

```
% Determines the threshold necessary for achieving 95% connectivity
% Normalized threshold is plotted vs. number of nodes
% Area = 100 micrometer x 100 micrometer


clc
clear all

set(0, 'defaultaxesfontsize', 14);
set(0, 'defaultlinelinewidth', 2);

Q=1000000;
ITER=10;

[x,y,e]=calc_node_trs_3d(100,1000,ITER,Q) ;

figure
errorbar(x,y/Q,e/Q,'-b')
hold on ;

xlabel('N') ;
ylabel('T/Q [\mum^{-3}]') ;
grid on ;
```

Number of nodes vs detection threshold for 100% connectivity in 3D

```
function [ trs_arr,nod_arr,err_arr ] = calc_node_trs_3d_100(
first,last,rpt,par )
%This function calculates concentration threshold vs number of nodes
%in which there is more than 100% connectivity
%with a given number of emitted particles and number of repetition


trs_arr=[] ;
nod_arr=[] ;
err_arr=[] ;

incr=100 ;
total=((last-first)/incr)+1 ;

R=zeros(rpt,total) ;
count=0 ;


k=(3*par)/(2*pi*exp(1)).^1.5 ;


   for n=first:incr:last

           count=count+1 ;

           for ll=1:rpt

               x=unifrnd(0,100,n,1) ;
```

```
 %generates random points in 100x100 area
              y=unifrnd(0,100,n,1) ;
              z=unifrnd(0,100,n,1) ;

              Rnd=[x y z] ;
              [d]=pdist(Rnd) ;  %measures the distance between nodes
              sq=squareform(d) ;


              trs=100 ;       %sets concentration threshold to 100
             adj=zeros(n) ;

              while trs>0
                    for ii=1:n      %constructs the adjacent matrix
                        for jj=1:n
                            if (k/(sq(ii,jj).^3)>=trs && ii~=jj);
                               adj(ii,jj)=1 ;
                            else
                               adj(ii,jj)=0 ;
                            end
                        end
                    end

                    spa=sparse(adj);

                   [NUM,COM]=graphconncomp(spa) ;
%finds the connected components of graph
                   unv = unique(COM);

                   rep = histc(COM,unv);
                   Mx= max(rep) ;
                   prc=(Mx/n)*100 ;
%finds the connectivity percentage

                   if prc==100
                      R(ll,count)=trs ;

                      break

                   else

                     trs=trs-1 ; %decreases 1 unit from
concentration threshold

                   end


               end


         end



    end
M=mean(R) ;
E=std(R)/sqrt(rpt) ;
```

```
cc=0 ;

  for  ii=first:incr:last
     cc=cc+1 ;

     trs_arr=[trs_arr;ii] ;
     nod_arr=[nod_arr;M(cc)] ;
     err_arr=[err_arr;E(cc)] ;

  end

end
```

```
% Determines the threshold necessary for achieving 100% connectivity
% Normalized threshold is plotted vs. number of nodes
% Area = 100 micrometer x 100 micrometer

clc
clear all

set(0, 'defaultaxesfontsize', 14);
set(0, 'defaultlinelinewidth', 2);

Q=1000000;
ITER=10;
[x,y,e]=calc_node_trs_3d_100(100,1000,ITER,Q) ;

figure
errorbar(x,y/Q,e/Q,'-b')
hold on ;


xlabel('N') ;
ylabel('T/Q [\mum^{-3}]') ;
grid on ;
```

Number of nodes vs detection threshold for 100% connectivity in 2D

```
function [ trs_arr,nod_arr,err_arr ] = calc_node_trs_100(
first,last,rpt,par )
%This function calculates concentration threshold vs number of nodes
%in which there is more than 100% connectivity
%with a given number of emitted particles and number of repetition


trs_arr=[] ;
nod_arr=[] ;
err_arr=[] ;

incr=100 ;
total=((last-first)/incr)+1 ;
```

```
R=zeros(rpt,total) ;
count=0 ;


k=par/(pi*exp(1)) ;


   for n=first:incr:last

          count=count+1 ;

          for ll=1:rpt

              x=unifrnd(0,100,n,1) ;  %generates random points in
100x100 area
              y=unifrnd(0,100,n,1) ;
              Rnd=[x y] ;
              [d]=pdist(Rnd) ;    %measures the distance between
nodes
              sq=squareform(d) ;


              trs=100 ;          %sets concentration threshold to
100
             adj=zeros(n) ;

              while trs>0
                    for ii=1:n      %constructs the adjacent matrix
                        for jj=1:n
                            if (k/(sq(ii,jj).^2)>=trs && ii~=jj);
                               adj(ii,jj)=1 ;
                            else
                               adj(ii,jj)=0 ;
                            end
                        end
                    end
                    spa=sparse(adj);

                   [NUM,COM]=graphconncomp(spa) ;
%finds the connected components of graph
                   unv = unique(COM);
                   rep = histc(COM,unv);
                   Mx= max(rep) ;
                   prc=(Mx/n)*100 ;
%finds the connectivity percentage

                   if prc==100
                      R(ll,count)=trs ;

                      break

                   else

                     trs=trs-1 ;
 %decreases 1 unit from concentration threshold

                   end
```

```
                end

            end


        end

M=mean(R) ;
E=std(R)/sqrt(rpt) ;


cc=0 ;

  for  ii=first:incr:last
     cc=cc+1 ;

     trs_arr=[trs_arr;ii] ;
     nod_arr=[nod_arr;M(cc)] ;
     err_arr=[err_arr;E(cc)] ;

  end

end



% Determines the threshold necessary for achieving 100% connectivity
% Normalized threshold is plotted vs. number of nodes
% Area = 100 micrometer x 100 micrometer

clc
clear all

set(0, 'defaultaxesfontsize', 14);
set(0, 'defaultlinelinewidth', 2);

Q=10000;
ITER=10;
[x,y,e]=calc_node_trs_100(100,1000,ITER,Q) ;

figure
errorbar(x,y/Q,e/Q,'-b')
hold on ;


xlabel('N') ;
ylabel('T/Q [\mum^{-2}]') ;
grid on ;
```