ANALYSIS OF RECEPTION PROCESS FOR AN ABSORBING RECEIVER IN MOLECULAR COMMUNICATION VIA DIFFUSION

by

Ali Akkaya

B.S., Computer Engineering, Boğaziçi University, 2003

M.S., Computer Engineering, Boğaziçi University, 2008

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Computer Engineering

Boğaziçi University

2015

ACKNOWLEDGEMENTS

First of all, I would like to thank my thesis supervisor Prof. Tuna Tuğcu not only for his guidance during this thesis but also for his endless support, kindness, and patience. His helpful comments and brilliant ideas made this possible.

I also would like to express my gratitude to Prof. Cem Ersoy, Assoc. Prof. Ali Emre Pusane, Prof. Sema F. Oktuğ and Assoc. Prof. Haluk O. Bingöl for their insightful and leading comments in my progress presentations and for being in my thesis committee.

Without the tremendous insight given by the nanonetworking research group, and friendship of all NETLAB members, it would be much more challenging to finish this thesis. In this regards, I would like to thank former and current members of the group Cem, Birkan, Gaye, Şükrü, Fadik, Salim, Tuğrul, Hakan, Gürkan, and all others for their accompaniment and friendship.

I would like to thank my friend Emrah. He inspired me with his academic path and I always feel his support even when there is long distance between us. I also thank Güler, Çınar, Ünal and all others with whom I feel calm, refreshed and happy.

Finally, I am really thankful to my girlfriend Dovile and my family. Dovile always made me feel special by caring, being supportive and bearing me in my ups and downs. My mother, brother, Selver and new member of our family, Can, are the sources of my joy in life. They were always next to me with their love and support.

This thesis is partly supported by State Planning Organization (DPT) of Republic of Turkey under the project TAM with the project number 2007K120610, Bogazici University Research Fund (BAP) under grant number 7436, and by Scientic and Technical Research Council of Turkey (TUBITAK) under Grant number 112E011.

ABSTRACT

ANALYSIS OF RECEPTION PROCESS FOR AN ABSORBING RECEIVER IN MOLECULAR COMMUNICATION VIA DIFFUSION

Nanotechnology is currently being applied to vast number of fields to overcome the challenges faced with existing technologies that cannot efficiently scale down to nano level. However, considering the limited processing and memory resources of nano-machines, performing complex tasks requires new communication mechanisms. Communication is one of the important issues to be addressed in nano-scale environment. Inspired by the nature, molecular communication via diffusion is a candidate to address this issue. Although the reception process of the messenger molecules has a significant impact on the performance of molecular communication via diffusion, the factors that effect the received signal for an absorbing receiver have not been investigated in the literature. In this thesis, we first introduce methods for efficient simulation of molecular communication via diffusion to enable further analysis. We propose two novel simulation architectures; a dual-zone simulation model to decrease execution time while preserving simulation accuracy and an HLA based architecture for distributed simulation of molecular communication via diffusion. Then, we analyse different dimensions of reception process for an absorbing receiver to derive closed form formulations. The results presented enable optimizations that will have a direct effect on production costs of receptors and the receivers. Finally, we propose a new approach for demodulation of information for an absorbing receiver and analyse energy consumption and data rate for the proposed model.

ÖZET

DİFÜZYON İLE MOLEKÜLER HABERLEŞMEDE SOĞURAN ALICI İÇİN ALIM SÜRECİNİN ANALİZİ

Nanoteknoloji, mevcut teknolojilerin nano seviyede yetersiz kaldığı durumların üstesinden gelmek için birçok alanda kullanılmaktadır. Ancak, nano-makinelerin kısıtlı işlemci güçleri ve hafızaları düşünüldüğünde, karmaşık işlemlerin yapılabilmesi için yeni iletişim yöntemlerine ihtiyaç duyulacagı anlaşılmaktadır. Nano seviyede iletişim, ele alınması gereken önemli bir problem olarak ortaya çıkmaktadır. Doğadan esinlenen difüzyon ile moleküler haberleşme yöntemi bu problemin çözümünde kullanılabilecek yöntemlerden birisidir. Taşıyıcı moleküllerin alım sürecinin, difüzyon ile moleküler haberleşme başarımına önemli bir etkisi olmasına rağmen, soğuran alıcının aldığı sinyali etkileyen faktörler literatürde yeterli ölçüde araştırılmamıştır. Bu tezde, öncelikle analizlerimizde kullanılmak üzere difüzyon ile moleküler haberleşme benzetiminin daha etkin biçimde yapılabilmesi için iki yeni model önerilmektedir. Önerilen modellerden ilki, benzetim hassasiyetini koruyarak daha hızlı çalısmasını sağlayan iki-alanlı benzetim modeli, diğeri ise difüzyon ile moleküler haberleşme benzetimlerinin dağıtık yapılmasına olanak veren HLA tabanlı mimaridir. Sonrasında, analitik formüllerin elde edilebilmesi için soğuran alıcı alım süreci değişik yönleriyle analiz edilmektedir. Sonuçlar, alıcı ve reseptör üretim maliyetlerinin eniyilemesini mümkün kılmaktadır. Son olarak da, soğuran alıcılarda bilginin demodülasyonu için yeni bir yaklaşım önerilmekte ve önerilen modelin enerji ihtiyacı ve veri hızı incelenmektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A	Total surface area of the receiver
A_α	Ratio of the surface area of the receiver to be covered by the receptors to achieve $F_{\text{hit}}^{r_s,n}(t) = \alpha$
$A^{r_s,n}$	Surface area of the receiver that is covered by the receptors
c_v	The capacity of a vesicle in terms of the messenger molecules
C	Concentration difference
$C(x,t)$	Concentration at point x time t
D	Diffusion coefficient
E_C	Energy required for carrying the secretory vesicles to the cell membrane
E_E	Energy required for releasing the molecules via the fusion of the vesicle and the cell membrane
E_S	Energy required for the synthesis of the messenger molecules from their building blocks
E_T	The total energy consumption to release molecules of a vesicle
E_{TOTAL}	The total energy consumption for the full communication
E_V	Energy required for the production of a secretory vesicle
$f_{\text{hit}}(t)$	Hitting rate of the molecules to the receiver at time t
$F_{\text{hit}}(t)$	The fraction of molecules absorbed as a function of time for a spherical absorbing receiver
$F_{\text{hit}}(t_1,t_1)$	The fraction of molecules received between t_1 and t_2
$F_{\text{hit}}^{r_s,n}(t)$	The fraction of molecules absorbed as a function of time for a spherical absorbing receiver with receptors
$F_{\text{hit}}^{r_s,n}(t_1,t_2)$	The fraction of molecules absorbed between time t_1 and t_2 for a spherical absorbing receiver with receptors
I	Diffusion current for a sphere with absorbing receptors
I_r	Diffusion current for a perfectly absorbing sphere
J	Diffusive flux
k	The dimension of the environment
m	The number of molecules

m_{max}	Number of molecules in a vesicle
M	The number of messages sent at each simulation time step
n	Number of receptors
n_α	Minimum number of receptors needed to achieve $F_{\text{hit}}^{r_s,n}(t) = \alpha$
N^{Rx}	Number of molecules received
$N^{\mathrm{Rx}}(t_1, t_2)$	Number of molecules received between t_1 and t_2
N^{Tx}	Number of messenger molecules transmitted
$p(r, t\|r_0)$	Molecule distribution function at time t and distance r given the initial distance r_0
P_d	The probability that a molecule diffuses out of the simulation boundaries
P_{hit}	The probability that a particle release from a point transmitter will be absorbed at steady state
P_i	The probability of sending a vesicle at a specific t_s
P_r	The probability that a molecule is received by a nanomachine
P_{rb}	The probability that a molecule is rolled back due to a collusion
P_s	The probability that a molecule moves to another medium slice
P_t	The probability that a molecule is transmitted by a nanomachine
r_r	Radius of the receiver nanomachine
r_s	Radius of the receptors
r_0	Transmitter distance to the centre of the receiver
r_{z_0}	The radius of $Zone_0$
$\overrightarrow{r}$	Total displacement in 3-D environment
R	Diffusion resistance for a sphere with absorbing receptors
R_r	Diffusion resistance for a perfectly absorbing sphere
S	Speedup achieved in simulation execution time
S_{dz}	Speedup achieved in simulation execution time for dual-zone model
T_{dz}	The execution time with dual-zone model
T_m	Execution time with multiple federates

T_s	Execution time with a single federate
T_{sz}	The execution time with a single zone
w	The rate of reaction with the receiver
$Zone_0$	The inner zone in the dual-zone simulation model
$Zone_1$	The outer zone in the dual-zone simulation model
Δt	Time step used in simulation of molecular communication
Δt_0	Time step used in $Zone_0$
Δt_1	Time step used in $Zone_1$
∇^2	Laplacian operator

LIST OF ACRONYMS/ABBREVIATIONS

BSD	Berkeley Software Distribution
BZD	Benzodiazepine
CNT	Carbon Nano Tube
DNA	DeoxyriboNucleic Acid
DoD	Department of Defense
FOM	Federate Object Model
GABA	γ-Aminobutyric Acid
HLA	High Level Architecture
IEEE	Institute of Electrical and Electronics Engineers
MCvD	Molecular Communication via Diffusion
MUCIN	MolecUlar CommunicatIoN
NS-2	Network Simulator 2
NanoNS	Nano Network Simulator
OMT	Object Model Template
RMSE	Root Mean Square Error
RTI	Run Time Infrastructure
TDMA	Time Division Multiple Access

1. INTRODUCTION

As the need for computational resources increases, there is a parallel increase in research activities to design smaller units with higher processing power. Nanotechnology is an outcome of these activities. One of the focuses of Nanotechnology is to construct functional units at nano-scale which can perform simple tasks. These functional units are generally referred to as nano-machines. They interact with each other to accomplish more complex tasks. For development of these functional units, three main approaches are followed, top-down, bottom-up, and hybrid3] [2].

- Top -down: In the top down approach, the aim is to scale down current electronic elements, which are at the micro-scale, to the nano-scale. There are several challenges faced in this approach since the physical laws governing the nano-scale are not the same with the micro-scale [3].
- Bottom-up: In this approach, new nano-machines are constructed from molecular components through chemical molecular reactions.
- Hybrid: In the hybrid approach, biological components are used as building blocks for nano-machines. Existing biological components are altered and combined to develop more complex systems.

1.1. Nano-networking

Considering the sizes of the nano-machines, it is expected that they will be able to accomplish computing of simple tasks. For nano-machines to perform complex tasks, they need to communicate with the external systems and among themselves. Nano-networking is the branch of science focused on communication between nano-machines to enable collaboration for complex tasks. The research activities on nano-networking can be grouped in four as electromagnetic, acoustic, nano-mechanical, and molecular [2]. Figure 1.1 shows this grouping.

Current microelectronic devices use electromagnetic waves for communication

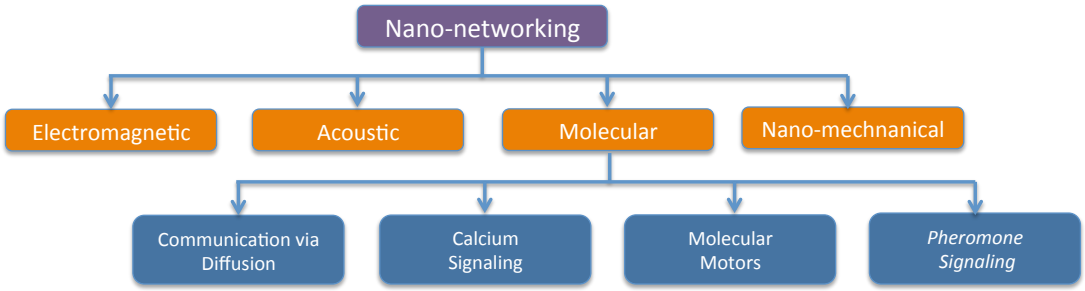


Figure 1.1. Nano-networking.

purposes, and at the nano-scale, electromagnetic waves are also one of the possible options for communication. Due to the small scale, only wireless option should be considered and this requires a nano-machine to include a radio transceiver operating at the terahertz band. In [4], the authors claim to realize this by integrating carbon nano tube (CNT) based antennas. Although this has been achieved, there is an important power limitation of nano-machines that makes creating a communication channel quite challenging [5]. In [6], the authors mention molecular absorption noise that is specific to nano-scale. Molecular absorption noise causes considerable path loss which allows communication at very small distances.

Acoustic communication, which uses ultrasonic waves for communication, requires special transducers to be integrated into nano-machines. Nano-mechanical communication, on the other hand, does not require a complex component in the nano-machine, but it assumes a physical connection between two nano-machines. These two approaches have not received much attention from the research community mainly due to these limitations.

Molecular communication became the most popular approach at nano-scale communication over the years, mainly because it is more promising since it is inspired by the nature. Researchers try to understand the mechanisms that are already used by living cells and their communication, and develop models accordingly. Among possible molecular communication options, Molecular Communication via Diffusion (MCvD) is the most prominent one since it is the core dynamic of most of other molecular

communication phenomena (discussed at Section 1.2).

Calcium signalling works using the fluctuations of the concentration of Calcium ions (Ca^{2+}). There should be a physical contact between two entities and gap junctions are formed for the Ca^{2+} flow. In [7], the authors give an overview of calcium signalling and discuss possible applications. Relaying possibilities for calcium signalling are investigated in [8]. In [9], the authors investigate the channel capacity of calcium signalling system based on an inter-cellular calcium wave model for astrocytes. As noted, the main limitation is that the nano-machines need to be physically connected.

Communication using molecular motors is a way of intra cell communication. Molecular motors move along microtubulus that are created within a cell to maintain its structure. Since molecular motors can only move on a specific path, this can be considered analogous to the wired communication. Usage of molecular motors for communication purposes is firstly introduced in [10].

Pheromones are the signalling molecules used in the nature by species for long range communication. It is introduced for possible nano-scale usage firstly in [11]. Pheromone signalling can be used to transmit the information gathered at the nano-scale to the macro-scale or to broadcast an information to a set of nano-machines.

1.2. Molecular Communication via Diffusion

Molecular communication is used by many living organisms to enable biological components to communicate among the cells of the organism, and is one of the methods that can be used for inter nano-machine communication. Molecular communication is the class of communication methods in which molecules are used for propagation of information. One of the ways for this propagation is diffusion, which is the basis of Molecular Communication via Diffusion. MCvD is a short-to-medium range molecular communication technique in which the messenger molecules diffuse in the propagation medium to transfer the intended information [12]. Similar to classical communication methods, communication via diffusion also has five main phases;

modulation, transmission, propagation, reception, and demodulation. The molecular communication model includes a transmitter nano-machine, a receiver nano-machine, and a propagation medium. After the information is modulated into molecules, the molecules are released to the medium by the transmitter. The molecules then propagate in the medium and arrive at the receiver. The following steps are repeated within each symbol duration. This approach is illustrated in Figure 1.2.

- *Modulation*: A vesicle that contains a group of molecules is prepared before each symbol duration.
- *Transmission*: At the start of the symbol duration, the molecules are released to the propagation medium.
- *Propagation*: The molecules diffuse through the propagation medium.
- *Reception*: Throughout the symbol duration, the molecules that reach the receptors of the receiver nano-machine are successfully received.
- *Demodulation*: The received molecular signal is demodulated to retrieve the modulated information.

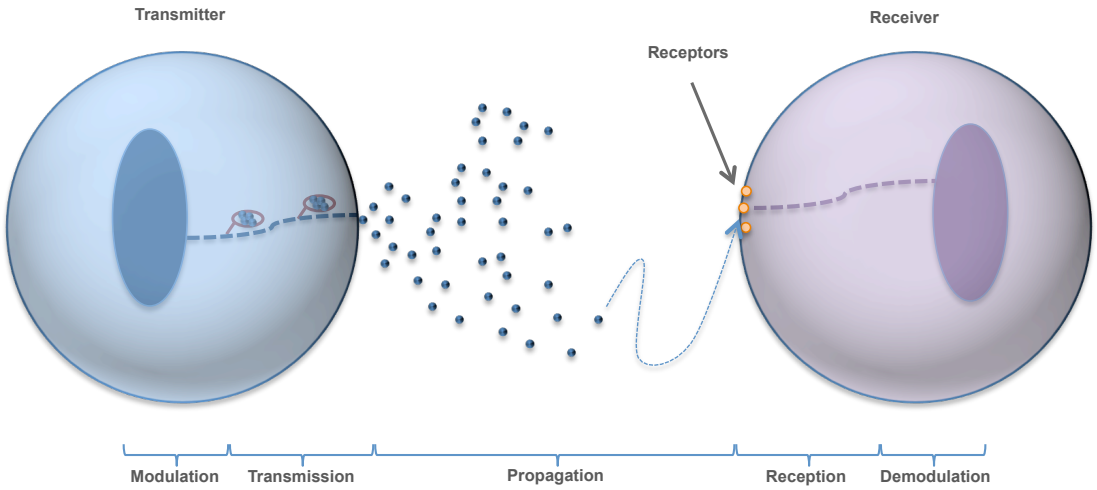


Figure 1.2. Molecular Communication System.

Diffusion, as a movement model for small particles, has been studied extensively in the 19^{th} and early 20^{th} century by scientists like Thomas Graham and Adolf Fick [13]. Diffusion focuses on capturing the general behaviour of a huge number of diffusing small particles in a medium (e.g., how a drop of dye diffuses through a body of water).

However, this diffusing behaviour is actually the macroscopic result of some basic movement that is conducted in the microscopic scale, called the Brownian motion. In other words, the movement of individual small particles can be modelled by Brownian motion dynamics; diffusion is the consequence of Brownian motion to reflect the group behaviour at the macroscopic scale. Molecular Communication via Diffusion research focuses on analytical and simulation based analysis of diffusion dynamics to create an effective communication channel among nano-machines.

1.3. Key Contributions

The contributions of this thesis are three fold; novel approaches for the simulation of molecular communication via diffusion, analytical model for an absorbing receiver with receptors, and alternative demodulation model for an absorbing receiver.

The contributions on simulation of molecular communication via diffusion can be summarized as:

(i) A dual-zone approach for simulation of molecular communication via diffusion to speed up the execution while preserving accuracy.
(ii) A distributed architecture based on High Level Architecture (HLA) that can be used to simulate complex scenarios. This architecture enables inter-operable and reusable simulation components and can be run in a distributed environment.

The contributions on the reception process for an absorbing receiver with receptors can be listed as:

(i) A realistic model to analyse the hitting rate for an absorbing receiver that considers receptors.
(ii) Analysis of the effect of receptor size and density on signal reception.
(iii) Design guidelines to optimize the production costs for receptors and receivers.

The contributions on demodulation options for absorbing receiver can be listed

as:

(i) An analog model as an alternative to digital modulation.

(ii) Energy consumption and data rate analysis for the proposed analog model.

1.4. Thesis Outline

The organization of this thesis is as follows: Chapter 2 includes information about related work on the topic. The architectures we propose for simulation of molecular communication are discussed in Chapter 3. The model defined to analyze the reception process for absorbing receiver in molecular communication is given in Chapter 4. Chapter 5 investigates the effect of absorbing receptors that are deployed over a receiver. Chapter 6 describes an alternative analog demodulation option for an absorbing receiver and analyzes the energy consumption and data rate. Finally, we conclude with Chapter 7.

2. RELATED WORK

There have been many research efforts about molecular communication in the literature. In this chapter, we detail relevant research activities. We start with works on broader topic of molecular communication. Afterwords, we focus on works about simulation models for molecular communication. Finally we present research activities investigating absorbing receivers.

2.1. Molecular Communication

In [14], molecular communication is introduced and modelled as five steps similar to the classical communication models, modulation, transmission, propagation, reception, and demodulation. When the transmitter nano-machine has information to be sent to receiver nano-machine, it follows proposed model to send messenger molecules to the environment, which propagate in the environment according to Brownian motion and are absorbed by receiver. Several research challenges are explored in the paper. One of the listed challenges is designing algorithms for modulation and demodulation of information in a robust manner.

In [2], Akyildiz *et al.* first define the nano-machine and its architecture, and provide a survey on different methodologies on nano-network communication for both short range and long range communications. They detail three common approaches for nano-machine development, top down, bottom up, and hybrid. In the top down approach, the target is to scale down the current micro-electro-mechanical technologies to develop nano-scale machines. In the bottom up approach, the idea is to use molecules as building blocks of nano-machines. In the hybrid approach, existing biological components are used to create new nano-machines. They also list possible applications of nano-networks, classifying them under four main categories: biomedical applications, industrial and consumer goods applications, military applications, and environmental applications.

In [15], a model of molecular communication based on gap junction channels is introduced together with a mathematical model. In the model, calcium signalling is used to encode information, which is then transferred from one cell to another via gap junctions. The usage of the model to solve current network design problems such as filtering and switching are also provided in detail. Selectivity and permeability of gap junction channels are identified as means for implementing filtering and switching functionality of a communication model.

In [16], an important characteristics of a communication model, channel capacity, is investigated for molecular communication. An approach based on information theory is used to define molecule delivery capacity of a molecular channel between two nano-machines and a closed form expression for capacity is derived. The closed form expression enables optimizing the channel parameters like temperature, concentration of medium, etc. to maximize the capacity. In [17], unicast, broadcast, and multicast capacity expressions are derived. The authors suggest the use of different types of molecules if two channels are places close to each other. This makes sure that the messenger molecules do not interfere with each other.

[18] and [19] also investigate the capacity of molecular communication. In the first paper, it is shown that the number of molecules reaching at the receiver is exponentially decreasing with distance between the transmitter and the receiver nano-machines. In the latter paper, it is shown that the channel in communication via diffusion shows a linear behaviour, which does not change over time.

In [20], energy model of a molecular communication model is developed to show how channel capacity can be optimized with energy limitations. In the model, energy used in production of messenger molecules, transportation of the messenger molecules within the transmitter nano-machine, and release of the messenger molecules from the nano-machine are considered. The effect of energy limitations on channel capacity is also investigated. In [21], Kuran *et al.* consider a cylindrical tunnel between spherical transmitter - receiver pair and analyse the effect on communication.

In another work [22], Kuran *et al.* consider a topology with two communicating transmitter-receiver pairs and investigate concentration shift keying and molecular shift keying It is shown that using molecular shift keying, the communication will be less interfered by the nearby communications.

A survey on molecular communication opportunities and challenges are listed in [23]. The architecture, features, applications, design, engineering, and physical modeling of molecular communication are considered. Similarly, [24] includes a comprehensive survey on recent enhancements in molecular communication domain.

2.2. Simulation of Molecular Communication

Simulation plays an important role in molecular communication research. Most of the previous research benefit from simulations for analysis of molecular communication models. Besides the use of simulation as a research tool, there are several works that have been performed specifically on nano-scale simulation design.

[25] is the first work mentioning the need for the simulation of molecular communication. The paper briefly describes the simulation requirements of a molecular motor based communication network.

In [26], a simulator for 3-D Brownian motion is proposed for a plane receiver. The simulator is capable of modelling nano particles under various configurable circumstances to simulate molecule diffusion and reception. The paper proposes a dual time step approach to cope with the run time complexity of a high number of particles. When the particle is far from the target, the movement is simulated in large time steps, and when it is closer to the target, smaller time steps are used. The dual-zone model we propose (Section 3.2) extends this approach for a point transmitter and a spherical receiver.

In [27], the authors introduce a C++ and Tcl based simulation framework (NanoNS) developed on top of the commonly used NS-2 discrete event simulator targeted for net-

working research. It implements diffusive molecular communication in 3-D space using a reaction-diffusion algorithm. The diffusion algorithm is based on the multi particle lattice gas automata algorithm in which the exact location of particles are not tracked but the medium is divided into lattice slides. Numerical analysis of the presented scenarios are used for the verification of the simulation framework, along with performance evaluation.

N3Sim [19] is a Java based simulation tool for diffusion based molecular communication. It enables the evaluation of molecular networks performance in 2-D and in 3-D space for specific scenarios. It uses Brownian motion and considers particle inertia and collisions among particles. The sensing of the local concentration is used for the reception model [28, 29].

In [30,31], a simulation platform for modeling information exchange at nano-scale is introduced. A Java-based software library is created using object oriented concepts. Elastic collision among molecules and receptor-based reception mechanism are implemented. A case study is used to demonstrate the features of the simulation tool. [32] introduces the simulation of diffusion-based molecular communications with drift inside blood vessels. It provides the description of the simulator and provides results for molecular signaling and communication potentials inside bounded spaces. The paper also discusses the execution of the simulator on a computational grid infrastructure. The same simulator architecture is also used to simulate in vitro experiments [33].

In [34], a simulation framework is introduced for simulation of neuron-based molecular communication. The proposed model integrates several components, and uses a TDMA-based signalling protocol case study to verify the model and analyse the performance.

A custom simulator for MCvD systems called MUCIN is presented in [35]. The proposed simulator is described in detail and it is mentioned that the source code is available under BSD licensing for contributors. The paper also presents a case study to analyse inter symbol interference mitigation and performs a performance evaluation

of the case study.

2.3. Receiving Process for Absorbing Receiver

In the literature, the attempt to define an analytical approach for nano-networks starts from one dimension [36–38]. These works include the first hitting time probability in closed form for one dimension.

For a spherical volumetric receiver, the channel response function for 3-D environment is provided in [39]. The model is also used to analyse utilizing enzymes in the propagation environment to mitigate inter symbol interference.

A spherical absorbing receiver in 3-D is first considered in [40]. Along with many other analytical analysis for diffusion dynamics, the authors provide the basis for the analysis of hitting probability of a particle released from a point transmitter to an spherical receiver.

[41] is the first paper providing and analysing the channel response functions for an absorbing spherical receiver in nano-networking by means of analytical manipulation of the diffusion equation. The work also includes peak time and energy analysis of the signal and comparison to the traditional communication methods.

The hitting probability for a spherical absorbing receiver with receptors at steady state is first analysed in [42]. An analogy with the electricity domain, where n conductive patches are located on an insulating sphere, is used to model an absorbing receiver. The insulating sphere in the model is analogous to the receiver and the receptors that bind with the molecules are analogous to the patches through which the current flows.

3. SIMULATION OF MOLECULAR COMMUNICATION

Research on channel characteristics, capacity, modulation schemes, propagation models, networking architectures, or performance evaluation of molecular communication heavily depends on simulation tools, either as the only means to analyse the proposed model, or to verify the analytical model. Molecular communication has its own characteristics, which are different from the traditional communication paradigms. Since it is based on the diffusion of molecules, the delay can only be predicted statistically, and possibility of chemical reactions in the environment creates a noise that needs to be modelled [22]. Current simulation tools that are developed for traditional communication models are not suitable to be used as they are for simulation of molecular communication. Extension of current simulation tools or development of new tools are necessary to support research groups working on molecular communication. The simulation model of communication via diffusion should cover the environment, the transmitter, propagation, and the receiver.

(i) *Environment:* The size of the simulated space is the first abstraction point while modelling the environment. If the real world scenario to be simulated is a confined space, same should be implemented on the simulator. For unconfined space simulations, a sufficiently large environment size can be used to approximate the real life scenario. This will keep the simulation executions shorter. The environment model should also include the parameters that affect the movement of the particles in the environment. Time step is another simulation parameter that affects the simulation accuracy and execution time. The simulation time step should be selected small enough to effectively model the real case scenario. Small time step may also increases the simulation durations to a level that is not acceptable. Time step is an important design parameter for molecular communication simulation and discussed in more details at Section 3.2.

(ii) *Transmitter:* The shape of the transmitter is the first design parameter for a simulator. In general for the simulations to justify theoretical work in which a point transmitter is used, similar model is used for the simulation. A spherical

surface can be used to model the transmitter for a better model of a real life scenario. In the nature, the cells may be of arbitrary shapes which can be reflected to the simulations. The computational cost of simulating such arbitrary shapes may be much higher. The researcher should decide on the abstraction level based on the research question at hand. We also need to decide on the number of transmission points over the transmitter, and the transmission period.

(iii) *Propagation:* We need to model the movement of the particles in the environment. The physical and chemical rules governing the environment should be analysed to model the simulation. We describe how Brownian motion can be used to model the diffusion at Section 3.1. For environments with drift or other particles that can have chemical interaction, other models need to be utilized.

(iv) *Receiver:* Similar to the transmitter, the shape is an important parameter. A spherical surface or an arbitrary surface can be used to model the receiver. The reception process is also critical. The receiver may be a non-absorbing in which messenger molecules may enter and exit the receiver. In general such model is used for the theoretical works. The receiver can be modelled as an absorbing receiver that absorbs the messenger molecules that hit the surface of the receiver. A more realistic model may also consider receptors on receiver surface which are used to absorb the messenger molecules.

Once the simulator design is completed, the simulation execution plan including all input and output sets to be simulated, and simulation scenario details should be created. The parameters used for the simulations should be sufficient for statistically meaningful results, while they should also be optimal for minimizing the simulation execution time. Both the simulator design, and the simulation execution plan are important steps of molecular communication research process as shown in Fig. 3.1.

Prior to each research project, a research team should consider the design issues for the simulator selection or development, and also carefully design the simulation execution plan. These two issues greatly influence the time to conclude any results out of simulation execution outputs. They also affect the evaluation of alternative options for the proposed model. Flexibility, re-usability, interoperability, and scalability should

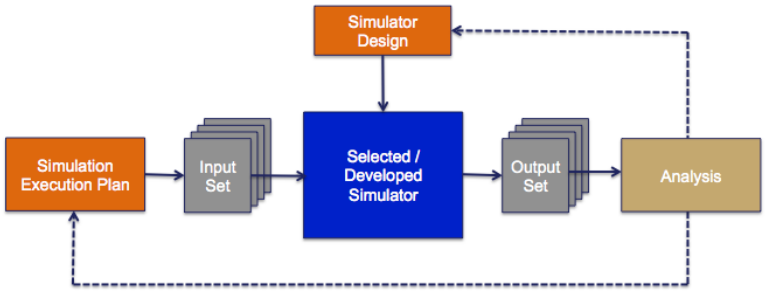


Figure 3.1. Simulation Process.

be considered during simulator design process. Based on the needs of the research project, an existing simulator can be utilized, or a new simulator can be developed. For the simulation execution plan, researchers should concentrate on the number of replications for Brownian motion randomization and the number of different input sequences used. These parameters affect the simulation time considerably, hence they need to be optimized to minimize simulation time, while still resulting in statistically meaningful outputs.

This chapter aims to guide the reader on the simulator design issues that will help to plan and execute the simulation process of molecular communication research activities. First, basics of the simulation of communication via diffusion is described. Then, a dual-zone based approach is introduced to optimize simulation execution time. Finally, a distributed architecture is described which supports software component reuse and high level of scalability. If such a common architecture can be agreed upon by different research groups, a growing library of simulation components can be developed, and complex scenarios can be simulated in a distributed environment.

3.1. Basic Simulation of Brownian Motion

Molecules are free to move in a fluid environment; thus, they move in a random fashion. We study the nature of this random motion within two perspectives [13]: Macroscopic and microscopic views. First, we focus on the macroscopic theory.

The macroscopic theory of diffusion can be developed from two simple and basic assumptions. The first of these is that a substance will move down its concentration gradient. Steeper gradient results in more movement of the material. If the relation between gradient and flux is linear, then in one dimension we have what is known as Fick's first law

$$J = -D\frac{\partial C(x,t)}{\partial x} \tag{3.1}$$

where x is the position, $C(x,t)$ is the concentration at that point, and D is the diffusion coefficient. The variable J is the flux and is defined as the amount of material passing across the point at x (or through a unit area perpendicular to the direction of flow) per unit time. The minus sign means that the flow is in the direction of decreasing concentration.

In a small element of length dx, the flux into the element from the left is different from the flux out of the element from the right. The difference between the two fluxes $J(x)$ and $J(x + \Delta x)$ determines how much material accumulates within the region bounded by x and $x + dx$ in a time interval Δt

$$(J(x + \Delta x) - J(x))\Delta t = -\Delta C \Delta x. \tag{3.2}$$

After rearranging and converting into derivative form, we get Fick's second law.

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2} \tag{3.3}$$

Equation 3.3 is for the one dimensional case. In three dimensions, the spatial derivative is replaced by the gradient, and combining with the second law we get

$$\frac{\partial C(x,t)}{\partial t} = D\nabla^2 C(x,t) \tag{3.4}$$

where ∇^2 is the Laplacian operator.

If we just consider the diffusion process starting from the origin, the concentration at x and time t is given by

$$C(x,t) = \frac{1}{(4\pi Dt)^{k/2}} e^{-||x||^2/4Dt} \tag{3.5}$$

where k and D are the dimension of the environment and the diffusion coefficient, respectively [43]. The value of D depends on the temperature of the environment, viscosity of the fluid, and the Stokes' radius of the molecule [44].

The microscopic theory of diffusion is utilized for simulating the motion of diffusing particles. The simulation of the diffusion is founded on simulating individual molecule movements according to the Brownian motion, rather than trying to come up with a closed form solution. The simulation technique for the Brownian motion is derived from the general solution of the Fick's diffusion equation.

In the one-dimensional space, the displacement of a single particle in unit time is a random variable ΔX, which follows a normal distribution with zero mean and σ^2 variance

$$\Delta X \sim \mathcal{N}(0, \sigma^2) \tag{3.6}$$

where $\sigma = \sqrt{2D\Delta t}$, and D is the diffusion coefficient that describes the tendency of the propagating molecules to diffuse through the fluid [42]. As an alternative option for simulating the Brownian motion, one may select the direction randomly and move same amount. Both schemes are equivalent when Δt is small. When the direction is selected randomly and a fixed length movement is used, the movement becomes correlated. Hence, having normally distributed step lengths have some advantages to simulate the continuous diffusion process. A sample implementation for the simulation of particles in one dimension using Java programming language can be found at Appendix A.

If the particles propagate through a three dimensional environment, this movement can be modelled as three independent displacements (one for each dimension) [45]

and the total displacement, $\overrightarrow{r}$, in one time step can be found as

$$\overrightarrow{r} = (\Delta x, \Delta y, \Delta z). \tag{3.7}$$

Sample implementations for the simulation of particles in two and three dimensions using Java programming language can be found at Appendix B and Appendix C respectively.

Selection of Δt is an important design decision. Smaller Δt provides better accuracy since the abstracted path of a molecule better approximates to the real path. Larger Δt results in less accurate simulation results, but provides faster simulation execution time. While determining Δt, one should take both the topology and the simulation scenario into account. In terms of topology, smaller receivers, receivers with arbitrary shapes (e.g. with a border with intricate details), or receivers with receptors require very small time steps in the order of nanoseconds, whereas for larger targets such as simple models of animal cells ($\sim 10\ \mu m$ diameter) a time step in the order of microseconds should be sufficient.

Particles (molecules in this scope) are assumed to have spherical bodies. Properties of the molecule and the environment determine the diffusion coefficient, hence the movement dynamics. Equation 3.6, suggests generating Gaussian random numbers with the given parameters for each dimension and at each step independently.

The movement at each time step should be created to simulate a particle movement starting from the origin without replication. In each time step, the movement in each dimension is normally distributed and the movement can be considered as the accumulation of normally distributed random variables. 75 milliseconds trace of the movements of a molecule is depicted in Figure 3.2.

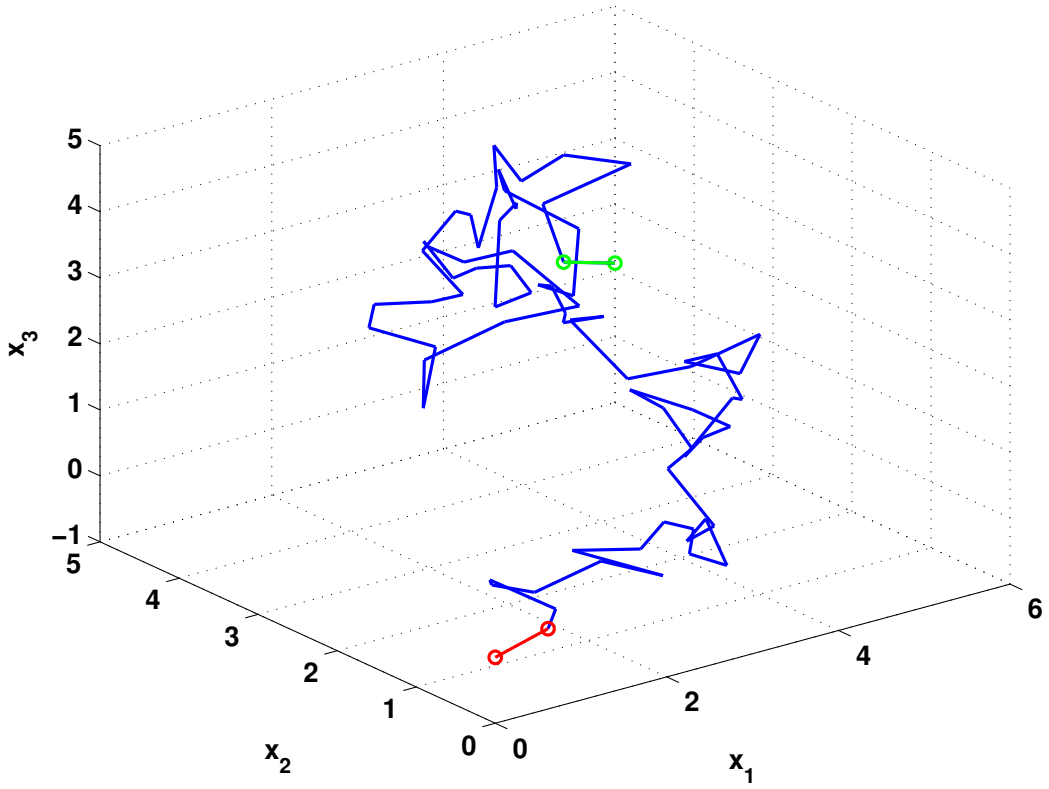


Figure 3.2. 75 miliseconds trace of a diffusing molecule.

3.2. Dual-zone Simulation of Molecular Communication

In a simulation environment, in addition to the arbitrary topology, we can model absorption or molecular degradation by simply removing the molecules from the environment depending on their location (i.e. remove the ones crossing the receiver border) or their life-time (we can select an arbitrary distribution and utilize a random variable with that distribution to decide when to remove a messenger molecule [46]). This gives extreme flexibility to the simulations. However, in order to effectively follow the restrictions imposed by the environment, the simulation time step should be selected very small (on the orders of microseconds or smaller, depending on the nature or the model of the receiver). Small time steps can increase the simulation durations to a level that is far beyond acceptable level for timely research results.

In Figure 3.3, we illustrate a simple case where the path of a messenger molecule

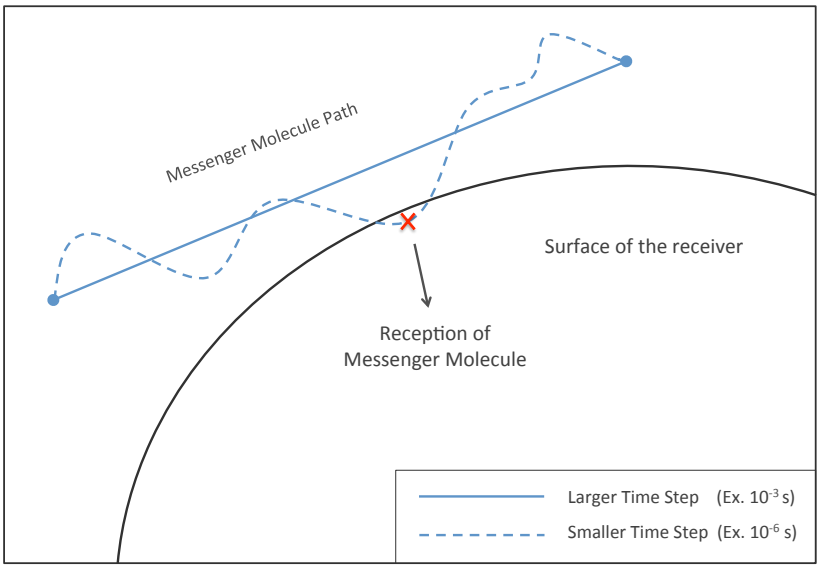


Figure 3.3. Effect of simulation time step on messenger molecule movement.

is depicted for large and small time steps. When larger time steps are used, the actual path the messenger molecule follows is coarsely represented. Therefore in simulations, reception of the messenger molecule at the receiver may be overlooked, resulting in reduced number of received molecules, consequently lower accuracy. On the other hand, using smaller time steps better represents the actual path of the messenger molecule and provides superior accuracy in exchange for longer simulation execution time. As a summary, larger time steps may cause failure in detecting reception of messenger molecule at the receiver. Smaller time steps provide finer grained simulation and better accuracy.

In this section, we first investigate the effect of time step selection on simulation accuracy and execution time in MCvD. Second, we propose a dual-zone model to decrease simulation execution time while keeping the simulations accurate. Our results indicate that the simulation time step has a significant effect on the accuracy of the simulation results. Moreover, with the introduction of dual-zone model, we achieve significant levels of accuracy in a simulation with up to 30 times faster execution times. The proposed dual-zone model is specifically beneficial when the effect of the input sequence length and distance is studied in detail.

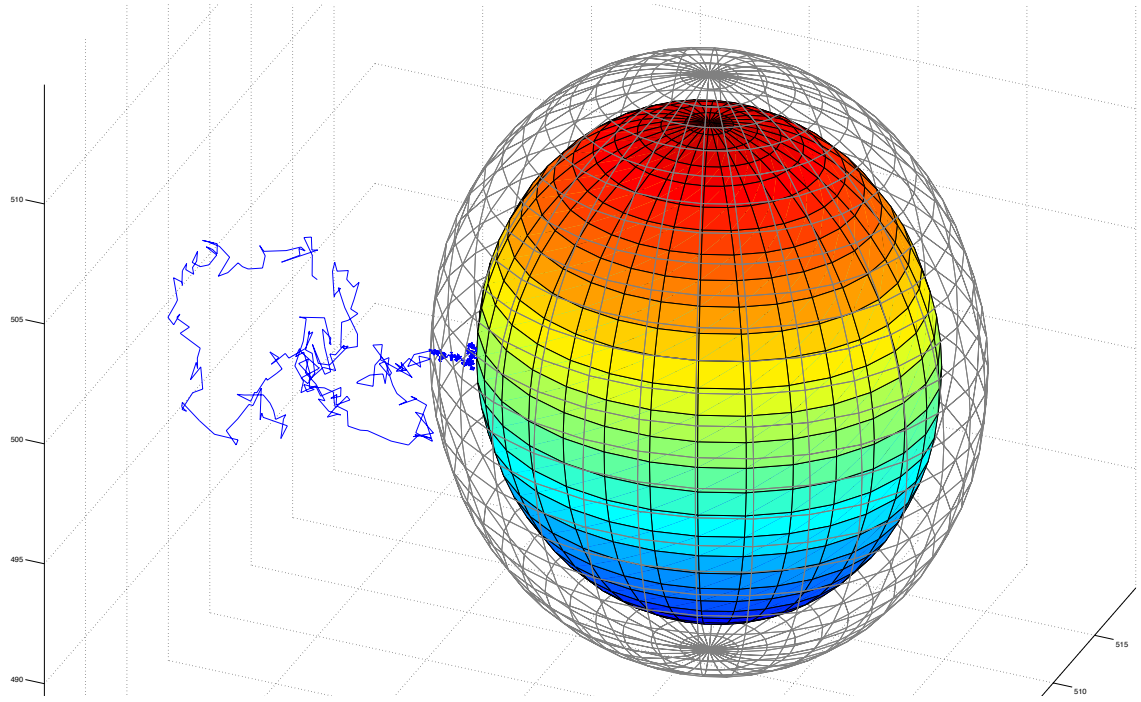


Figure 3.4. Dual-zone simulation model.

3.2.1. Dual Zone Simulation Model

The proposed dual-zone simulation model utilizes the dual time-step approach used in [47] and extends it for a point transmitter and a spherical receiver as illustrated in Figure 3.4. In the figure, an accurate representation (taken from the simulation) of the movement of a molecule according to the dual-zone Brownian motion is represented. A hypothetical sphere around the receiver separates the environment into two zones, namely $Zone_0$ and $Zone_1$. The molecules in $Zone_1$ use a larger time step, Δt_1. As soon as a molecule goes into $Zone_0$, it continues to move using a smaller time step defined for $Zone_0$, Δt_0. The outer zone offers faster execution with longer time step and coarse grained representation of the movement whereas the inner zone provides accuracy with a much smaller time step and a detailed model of the movement of the molecule. As stated in Section 3.1, the accuracy of the movement of a molecule highly depends on the selected Δt, which determines the displacement of the molecule in each simulation step. Using a larger time step for the molecules that are sufficiently far away from the receiver results in shorter execution times, and the smaller time step for the molecules closer to the receiver enables accurate simulation results. The model makes use of

zones to shorten simulation time while achieving high accuracy.

In our implementation, the movement of the molecule is abstracted by taking snapshots at each Δt, and the position of the molecule is calculated for each snapshot. The exact trajectory is achieved with the assumption that the molecule follows a direct path between these point. Δt used for the overall simulation equals to Δt_0 since $Zone_0$ has smaller time step than $Zone_1$ and the events scheduled for $Zone_0$ is more frequent. For each molecule, the next position is calculated using the time step of the zone that the molecule resides in, and the next location update is scheduled in Δt seconds depending on the zone. Molecules in $Zone_0$ perform much more frequent location updates, resulting in accurate results, and molecules in $Zone_1$ perform fewer location updates, resulting in faster execution. For instance, when Δt_1 is 1 ms, and Δt_0 is 1 μs, the simulation executes at $\Delta t = 1$ μs. The molecules in $Zone_0$ update their loci in every simulation step, while the molecules in $Zone_1$ perform location update in every 1000^{th} simulation step. The trajectory for a molecule in $Zone_1$ is abstracted as a straight line between two positions. When a molecule moves from $Zone_1$ to $Zone_0$, it starts to move in every simulation step, and when a molecule moves from $Zone_0$ to $Zone_1$ it starts to update its location in every 1000^{th} simulation step. It should be noted that the implementation should handle the transition cases carefully to make sure that simulation accuracy is not affected. For instance, when a molecule moves into $Zone_0$, its next move should only be scheduled at the end of the 1000^{th} step. Similarly, if all molecules in $Zone_1$ perform the moves concurrently, the molecules that move to $Zone_1$ have to wait for the next schedule, which results in incorrect behaviour. Since the way how the transitions are implemented may affect the accuracy of the simulation, special care should be given to implementation of transitions between zones.

In the proposed model, an important design parameter is the radius of $Zone_0$, r_{z_0}. It has a direct effect on the performance of the dual-zone simulation model. If this radius is not selected large enough, then the simulation will not be accurate sufficiently. If it is selected larger than required, then the simulation model will not benefit from using dual time step approach, and will not execute faster. To select r_{z_0}, we use the three sigma rule in statistics. The three-sigma rule is used to signify the range in

which the values of a normal distribution lie. According to this rule, 68.2% of the values in a normal distribution lie in the range $[\mu - \sigma, \mu, \mu + \sigma]$, 95.4% of the values in $[\mu - 2\sigma, \mu, \mu + 2\sigma]$ and 99.7% of the values in the range $[\mu - 3\sigma, \mu, \mu + 3\sigma]$. Since the movement of a molecule follows a Gaussian distribution with $\mu = 0$ and $\sigma = \sqrt{2D\Delta t}$ in each dimension, to make sure that a molecule in $Zone_1$ does not jump over $Zone_0$, we select r_{z_0} as

$$r_{z_0} = r_r + 3\sqrt{3}\sqrt{2D\Delta t_1}, \tag{3.8}$$

where r_r is the radius of the receiver and Δt_1 is the time step for $Zone_1$. This simulation model benefits from computational resources since for the molecules in $Zone_1$, it only calculates new molecule position at Δt_1 time intervals. A molecule will not be absorbed by the receiver without passing through $Zone_0$ with 99.1% probability when we use such a radius for $Zone_0$. Thus, the selected radius provides an accuracy comparable to that of a simulation which is executed using Δt_0. It should be noted that the selection of this parameter depends on the simulation requirements. If very high accuracy is not needed, smaller r_{z_0} values can be used, which speed up the execution significantly. For an analysis on the selection of r_{z_0} the reader should refer to Section 3.2.2.1.

3.2.2. Results

To compare the performance of our dual-zone model with the simulation without zones, first we illustrate how the dual-zone system improves simulation execution time while maintaining accuracy. Next, we give the effect of simulation parameters such as input sequence length, transmission power, and communication distance on the amount of improvement on simulation execution time. Simulation parameters used are listen in Table 3.1.

For the performance evaluation of the proposed model, we define the dual-zone

speedup S_{dz} as

$$S_{dz} = \frac{T_{sz}}{T_{dz}}, \tag{3.9}$$

where T_{sz} is the execution time with a single zone and T_{dz} is the execution time in dual-zone approach.

Table 3.1. Range of parameters used in the experiments.

Parameter	**Value**
Number of messenger molecules (N^{Tx})	10 000
Liquid viscosity	0.001 $kg/(s\,m)$
Temperature	310 °K
Diffusion coefficient (D)	79.4 $(\mu m)^2/s$
Receiver radius (r_r)	10 μm
$Zone_0$ radius (r_{z_0})	$r_r + 3\sqrt{3}\sigma$ μm
Transmitter distance to the center of the receiver (r_0)	$r_r + 4$ μm
Simulation time step ($\Delta t_{1,0}$)	$\{10^{-3}, 10^{-6}\}$ s
Simulation duration	1 s

3.2.2.1. Effect of Time Step on Received Signal and Execution Time. How coarse a simulation is determines the length of the simulation execution time as well as the quality and accuracy of the results. In order to assess qualitatively the simulation results, we take the analytical formula presented in Chapter 4 for a spherical absorber as the ground truth and compare the simulation results with the analytical formulation. The fraction of absorbed molecules as a function of time is found as

$$F_{\mathrm{hit}}(t) = \frac{r_r}{r_0}\ \mathrm{erfc}\left[\frac{r_0 - r_r}{\sqrt{4Dt}}\right], \tag{3.10}$$

where r_r and r_0 denote the radius of the receiver and the distance between transmitter and the center of receiver.

Our first simulation scenario is designed to represent a case where $N^{Tx} = 10\,000$ molecules are released from a point source at $r_0 = 14\ \mu m$, which are then absorbed by the receiver with $r_r = 10\ \mu m$. In this case, a decent amount of deviation is observed when we select $\Delta t = 10^{-3}\ s$, whilst with $\Delta t = 10^{-6}\ s$ time step the deviation is negligible. The values for Δt are selected as $10^{-3}\ s$ and $10^{-6}\ s$ since these are typical values we use for a perfect and imperfect absorbing receivers. Apparently, the actual selection depends on the abstraction level required for the simulation, speed up and accuracy needs.

Table 3.2. Effect of r_{z_0} on Speedup and RMSE ($N^{Tx} = 10\,000$).

Simulation Type	$\Delta t\ (s)$	$r_{z_0}\ (\mu m)$	**Speedup**	**RMSE (molecules)**
Single Zone	10^{-3}	-	757.52	139.11
Dual Zone	$10^{-3}, 10^{-6}$	$r_r + \sqrt{3}\sigma$	27.47	13.01
Dual Zone	$10^{-3}, 10^{-6}$	$r_r + 2\sqrt{3}\sigma$	12.39	4.28
Dual Zone	$10^{-3}, 10^{-6}$	$r_r + 3\sqrt{3}\sigma$	6,04	2.72
Single Zone	10^{-6}	-	1	2.24

Table 3.2 shows a set of scenarios where dual-zone and single-zone simulations are executed with different zone sizes. To compare the accuracy we use Root Mean Square Error (RMSE). RMSE is calculated using the results of the simulations and the analytical results obtained using (3.10). The results show that the single-zone simulation with large time step yields to a 1.4% error (139.11 molecules out of 10 000 molecules transmitted) compared to the analytical solution. When the dual-zone approach is introduced, this error drops significantly to 0.13% (13.01 molecules out of 10 000 molecules transmitted) for the smallest zone radius and gradually reduces the error to the best case scenario where the whole simulation topology is constructed with a single-zone with a small time step ($\Delta t = 10^{-6}\ s$). In this scenario, our favored zone size of $r_{z_0} = r_r + 3\sqrt{3}\sigma = 12.07\ \mu m$ achieves the remarkable accuracy of 0.027% difference (2.72 molecules out of 10 000 molecules transmitted) to the base case with 6 times shorter execution time. This achievement displays a fraction of the potential for dual-zoning.

3.2.2.2. Effect of Signal Power and Input Sequence Length. In most cases, the research problem at hand requires a broad consideration of the system parameters. In MCvD research, signal power and the input sequence length are two prime examples of these parameters through which the behaviour of the system is largely influenced.

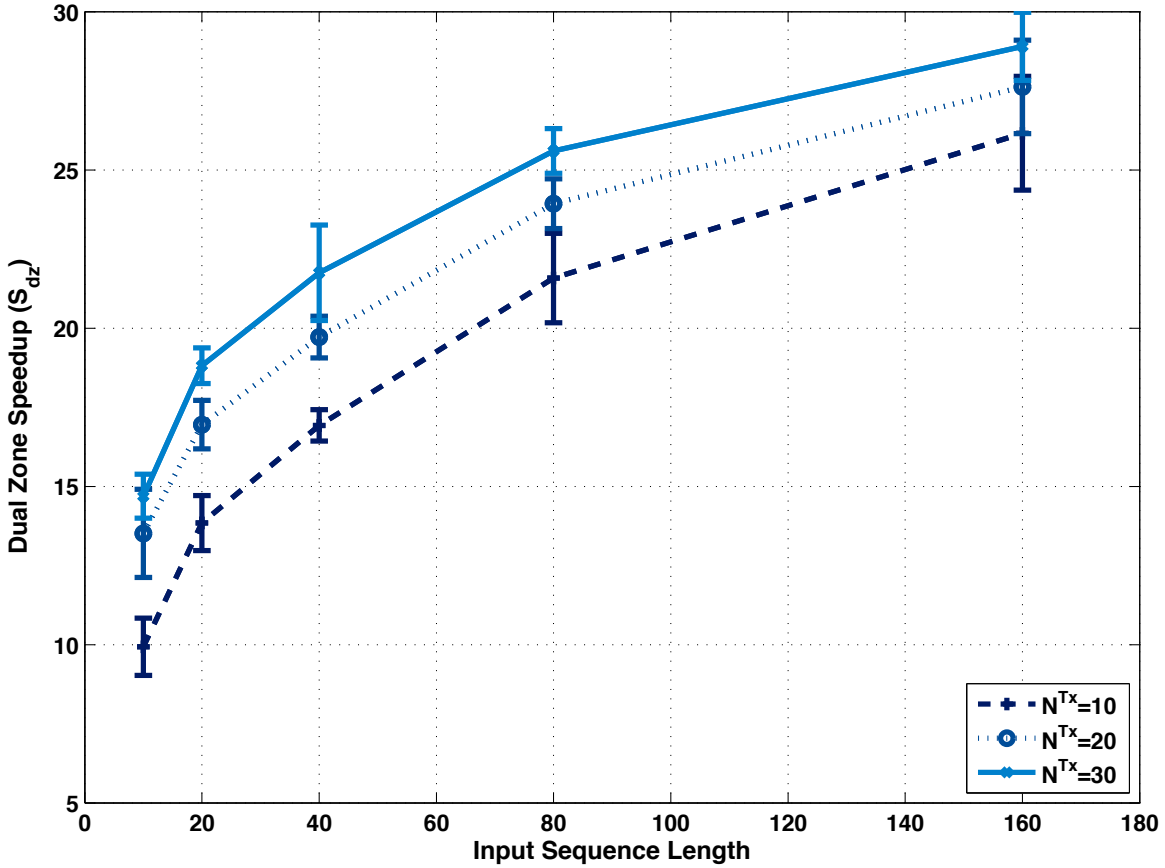


Figure 3.5. Dual Zone Speedup (S_{dz}) versus input sequence length for different N^{Tx} values.

In order to analyse the effects of these two parameters, we set up a set of simulations in which the transmission power and the input sequence length are altered using values from the sets $\{10, 20, 30\}$ and $\{10, 20, 40, 80, 160\}$, respectively. Figure 3.5 displays the speedup gained between the single-zone setup with $\Delta t = 10^{-6}$ s and the dual-zone setup with $r_{z_0} = r_r + 3\sqrt{3}\sigma = 12.07$ s, $\Delta t_0 = 10^{-6}$ s, and $\Delta t_1 = 10^{-3}$ s with 0.95 confidence interval.

Figure 3.5 reads the increase in the signal power also brings up a gain in the speedup for every input sequence length selected. The reason behind this gain is the increase in the number of accumulating molecules in the system. As the communication

progresses, the number of messenger molecules dwelling in the communication medium increases. For the simulation execution time, it matters if this dwelling occurs in $Zone_1$ where the simulation is more lightweight, or in $Zone_0$ where the simulation requires more computation. In the dual-zone case with the increase of N^{Tx}, the number of molecules dispersing into $Zone_1$ expectedly increases, since volume-wise $Zone_1$ is much larger than $Zone_0$. Consequently, the probability of being in $Zone_1$ is larger than being in $Zone_0$ where the gap between the respective probabilities also keeps widening as the time passes and the molecules keep dispersing.

Similarly, the increase in the input sequence length results in a higher speedup, where the rate of increase for S_{dz} decreases with increasing input sequence lengths. The reason behind the increase is the same with the signal power parameter. The molecule dispersion as the time passes favors $Zone_1$ instead of $Zone_0$, increasing the fraction of molecules that are in $Zone_1$ with time. The declining increase is explained with the decreasing rate of effective dispersion of the molecules. Expectedly, molecules from the earlier phases of the communication are already dispersed through the end of the communication and contribute minimally to the difference between the number of molecules in $Zone_1$ and $Zone_0$. As the input sequence length is increased we expect an increase in speedup with respect to input sequence length to converge to a value that is parametrized by N^{Tx}.

As a summary, the speedup increases with both the increase in the signal power and the input sequence length, since both entail an increase in the fraction of molecules that are dwelling in $Zone_1$, where the simulation execution is lightweight.

3.2.2.3. Effect of Communication Distance. Selecting a viable communication distance, which both fulfills system requirements and allows a healthy communication link to be established between nano-machines, is crucial. Therefore, distance selection is one of the most important design decisions in MCvD research. In order to investigate the effects of distance on speedup, we set up a simulation scenario with distances selected from the set $\{1, 2, 4, 8, 16\}$ with various sequence lengths.

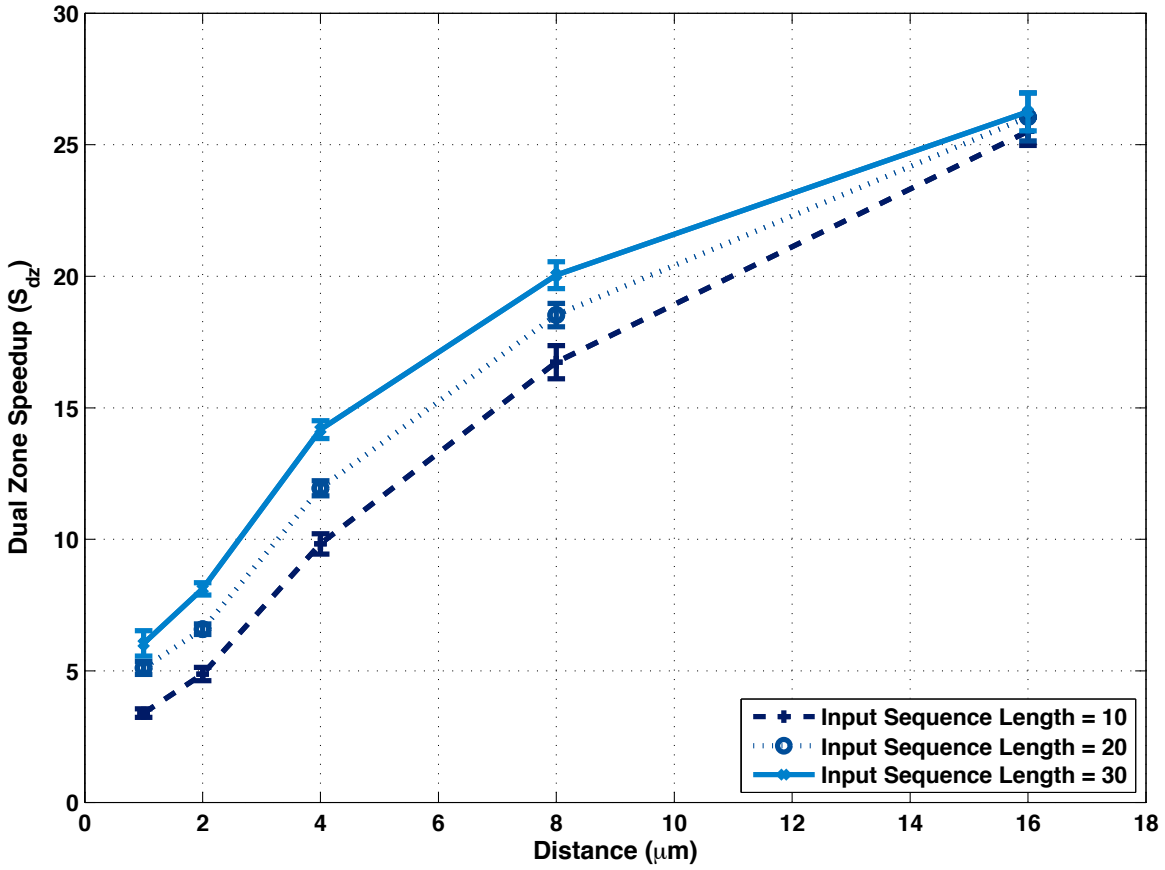


Figure 3.6. Dual Zone Speedup (S_{dz}) versus distance for different input sequence lengths.

Figure 3.6 displays the change in dual zone speedup, S_{dz}, with respect to both distance and the input sequence length with 0.95 confidence interval. From the figure we observe that the gained speedup increases in a declining rate as the transmitter and receiver pair is further separated. The reason behind the increase is, again, attributed to the distribution of molecules between the zones. At distance $d = 1\mu m$, we observe only a small increase in the speedup, since initially (i.e. moments after the release) most of the molecules are in $Zone_0$, where $r_0 = 11\ \mu m < r_{z_0} = 12.07\ \mu m$. These molecules contribute to speedup only after they randomly disperse to the $Zone_1$. In contrast, at distance $d = 16\ \mu m$ all molecules are initially in $Zone_1$ and they make the largest possible contribution to the speedup. Similarly, the effect of sequence length is minimized at large distances since the effect of dispersion over time becomes negligible.

Lastly, one must note that the speedup is directly related to the difference in

number of molecules dwelling in each zone. Here, the best case for speedup can be achieved in a hypothetical scenario where all released molecules dwell in $Zone_1$. Therefore, selecting longer distances yield to higher (eventually highest) speedup rates. In this respect, employing dual-zone approach in studies, where distance parameter is analysed, can be extremely beneficial in terms of simulation time.

As a summary, the speedup increases with both the increase in the input sequence length and the communication distance. The speedup gain is again associated with the distribution of molecules in each zone, which shifts in favour of the $Zone_1$ as the communication distance increases.

3.3. Distributed Simulation of Molecular Communication

In this section, we define a distributed simulation architecture [48] for molecular communication via diffusion based on the standardized High Level Architecture [49], which is widely used in large scale simulations. Since HLA only provides the basic infrastructure for distributed simulations, in our simulator, we propose a specific architecture for molecular communication, focusing on interoperability, re-usability, and scalability. We define the architecture and analyse the parameters that are affecting the parallelism and the possibility of load distribution in a distributed molecular communication simulation scenario. Using an HLA-based distributed simulator, it is possible to execute large simulation scenarios and interact with systems containing living organisms.

3.3.1. High Level Architecture

A distributed simulation is a collaborative system in which each simulation unit runs on an independent computational unit and communicates to simulate a scenario in a commonly managed logical time. HLA was developed by the United States Department of Defense (DoD) to facilitate the integration of distributed simulation models within a common architecture. The target was to cover defence applications but it has been used by many industry and research areas. It was then standardized by the

IEEE [49]. The main purpose of HLA is to define the component models and their interactions. These components are called federates, and they enable software reuse and abstraction. These components communicate over a Run Time Infrastructure (RTI) using standard protocols to form a simulation model referred to as federation. Each federation should have a Federation Object Model (FOM) that is created in accordance with an Object Model Template (OMT) defined by the standardization. This enables independent design and development of components and also distributed execution of the simulation. A generic HLA architecture is shown in Figure 3.7.

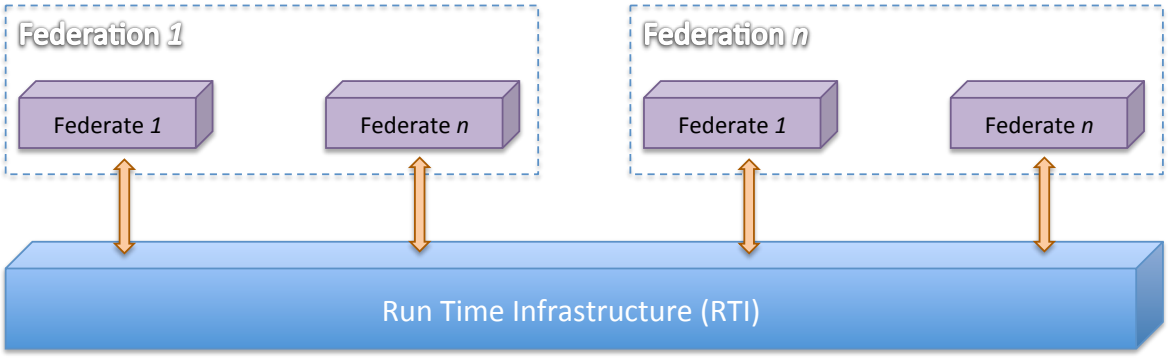


Figure 3.7. Generic HLA Architecture.

The RTI is the backbone of the federation, and provides synchronization, communication and data exchange services to the federates [50]. Each federate can be an independent event or time driven simulations, real time simulations with human interaction, live systems or equipments. The HLA does not restrict what is modeled in a federate, it defines the interaction among them. There are six classes of services that RTI provides [51]:

(i) *Federation management:* Basic functionality required to create and execute a federation.

(ii) *Declaration management*: Management of data exchange between federates, using the information provided by federates.

(iii) *Object management*: Creation, deletion, identification, and other services at the object level.

(iv) *Ownership management:* The dynamic transfer of ownership of object/attributes during an execution.

(v) *Time management:* Synchronization of runtime simulation data exchange.

(vi) *Data distribution management*: Routing of data among federates during federation execution.

The HLA interface specification defines the way these services are accessed, but how the services are actually implemented is not in the specification scope. Several RTI systems have been developed implementing the specification that can be used to develop HLA based simulation environment.

3.3.2. Distributed Simulation Architecture for Molecular Communication

Simulation tools are commonly used for analysis of molecular communication models. Each research group builds a separate tool from scratch, which makes simulating complex scenarios ineffective and sometimes impossible due to architectural or resource limitations. A common simulation architecture, which supports component reuse and distributed execution, will enable the creation of common component library to share knowledge, the efficient utilization of development efforts, and the definition and execution of large scale simulation systems. Research groups will benefit from the off-the-shelf components developed by other groups, and they will be able to contribute to the common library as well. Additionally, the architecture will enable cooperative execution of complex scenarios, hence the simulation infrastructures of different groups can be utilized to execute larger simulation scenarios. These benefits will boost the cooperation within and among research groups working on molecular communication.

3.3.2.1. Simulation Concept. At the conceptual level, HLA defines independent components called federates, and communication among them to create a federation. Thanks to the flexible component structure, it is possible to easily abstract the interaction with living organisms, other simulation systems, or passive observers who monitor the simulation execution. If the communication parameters of a living organism or laboratory on chip can be abstracted and interchanged in real time, they can be integrated into a more complex model where some parts of the model are experimental systems, and

some parts are simulation components. Decoupling and standardizing such integrations make feasible defining and executing such hybrid scenarios. In addition, decoupling using HLA standardization ensures that the components can be deployed to independent computation units and RTI handles all communication among the federates. This concept is depicted in Figure 3.8.

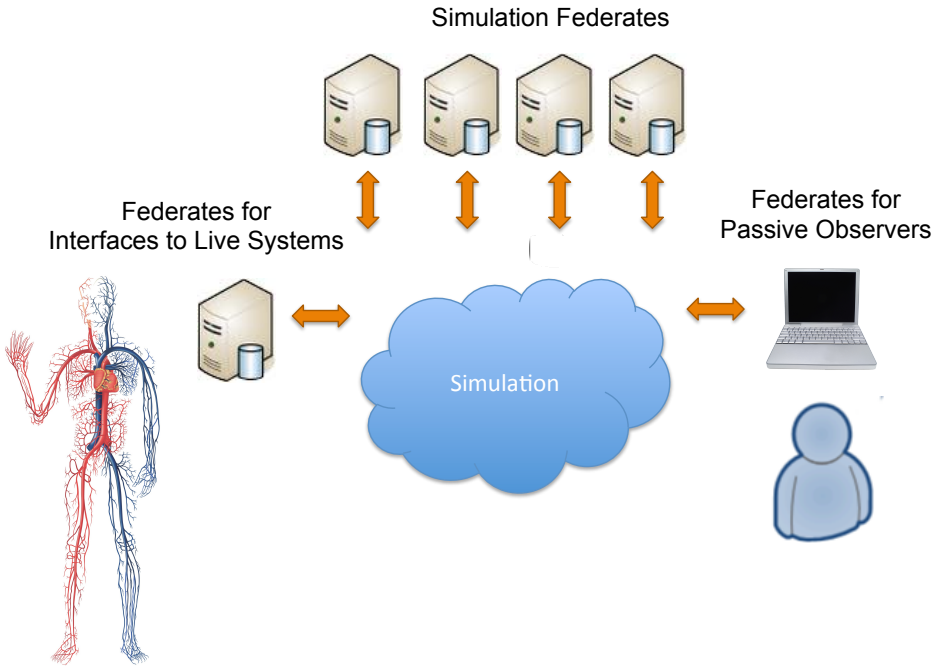


Figure 3.8. Simulation Deployment.

3.3.2.2. Simulation Design. We have interoperability, re-usability, and scalability as our main design criteria starting from the initial phase of our design. These criteria lead us to an architecture based on HLA, which addresses these concerns. Using HLA, it is possible for modules executing on different platforms to communicate. Common software libraries can be developed and used to create large scale simulations, and it is possible to run simulations in distributed computer systems. There are four primary principal benefits of executing a simulation program across multiple computers [52]:

(i) *Reduced execution time.* By dividing the large simulation computation into smaller sub-computations, and distributing these computational tasks to different computational units, it is possible to decrease execution time.

(ii) *Geographical distribution.* Running the simulation on geographically distributed computers enables interaction with users or living organisms, which can be in

different locations.

(iii) *Integrating simulators that execute on machines from different manufacturers.* Different research groups may want to develop components on different platforms, and also interaction with live systems may require the use of specific platforms. A distributed approach supports interaction between different platforms.

(iv) *Fault tolerance.* Using multiple computational units, it is possible to implement fault tolerance so that in case of a unit failure, another unit can pick up the work of the failing unit.

3.3.2.3. Simulation Components. The proposed architecture is shown in Figure 3.9. The simulation is defined in a Molecular Communication Federation. Separate federates are defined for molecules, nanomachines and the medium. The federates communicate with each other over RTI based on a common FOM. Portico [53], an open source, cross-platform HLA RTI implementation is used to develop the simulation tool. Portico uses another open source project, JGroups [54], for communication among federates. The communication among federates includes attribute updates and interactions. It is mainly dominated by number of molecules. The number of messages sent at each simulation time step can be formulated as

$$M = m \cdot (1 + P_t + P_r + P_d + P_{rb} + P_s) \tag{3.11}$$

where m is the number of molecules, P_t is the probability that the molecule is transmitted by a nanomachine, P_r is the probability that the molecule is received by a nanomachine, P_d is the probability that the molecule is diffused out of the simulation boundaries and destroyed, P_{rb} is the probability that the molecule is rolled back due to a collision, and P_s is the probability that the molecule moves to another medium slice since an additional message is transmitted for each one of these events. The upper bound for the number of messages is $2 \cdot m$, since the sum of P_t, P_r, P_d, P_{rb}, and P_s is bounded by one. Therefore, we can conclude that M is between m and $2m$.

The scalability is achieved by distributing molecule related tasks to Molecule

Federates, nanomachine related tasks to Nanomachine Federates, and slicing the 3-D space and assigning a medium federate to manage each slice. Each federate is an independent component and thanks to cross-platform support of Portico, each can be implemented using Java or C++ programming languages, and can be executed on Windows or Unix based systems. We use Java for our implementations, and Unix for simulation execution.

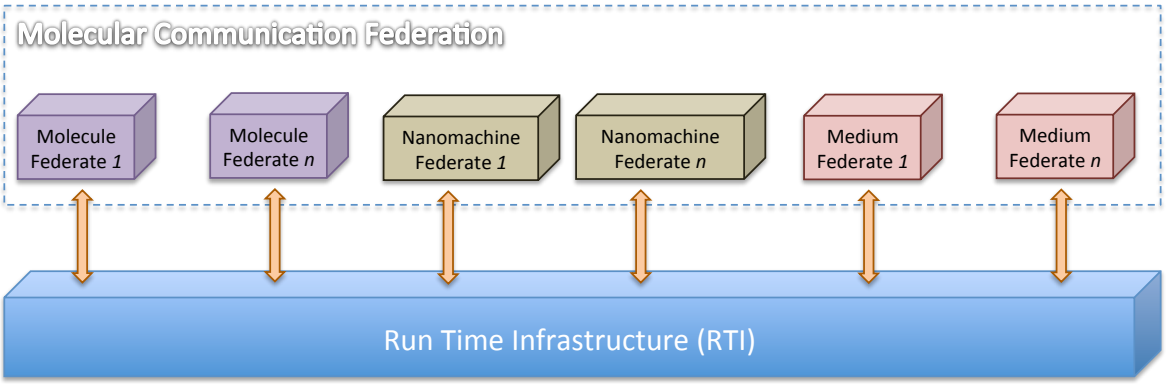


Figure 3.9. Simulation Architecture.

(i) *Molecule Federate:* The Molecule Federate abstracts the molecules in the simulation. It is responsible for molecule movements in the medium. It subscribes to the Medium Federate attributes, meaning that if the attributes of the Medium Federate change, this information is communicated to the Molecule Federate by RTI. The Molecule Federate calculates the next position of the molecules and updates the attributes accordingly. Currently an implementation for Brownian motion [55] is available, but it is possible to implement different diffusion laws. The molecules propagate through 3-D space, and this movement is modelled as three independent displacements, one for each dimension in the 3-D space. The total displacement $\vec{r}$ in one time step can be found as

$$\vec{r} = (\Delta x, \Delta y, \Delta z). \tag{3.12}$$

In each dimension of the 3-D space, the displacement of a molecule in one time step is a random variable, which has a normal distribution

$$\Delta x \sim \mathcal{N}(\mu, \sigma^2), \Delta y \sim \mathcal{N}(\mu, \sigma^2), \Delta z \sim \mathcal{N}(\mu, \sigma^2) \tag{3.13}$$

where μ is taken as 0, and σ is calculated as $\sigma = \sqrt{2D\Delta t}$, where D is the diffusion coefficient.

These environmental coefficients are defined as part of medium slices, and once a molecule moves into a slice, it starts moving according to the new coefficients. The published and subscribed attributes and interactions defined for the molecule object in FOM are listed below. The published attributes and interactions are communicated to other federates which subscribe to these attributes or interactions. This relation is shown in Figure 3.10 and the pseudo-code for the Molecule Federate is given in Algorithm 3.11.

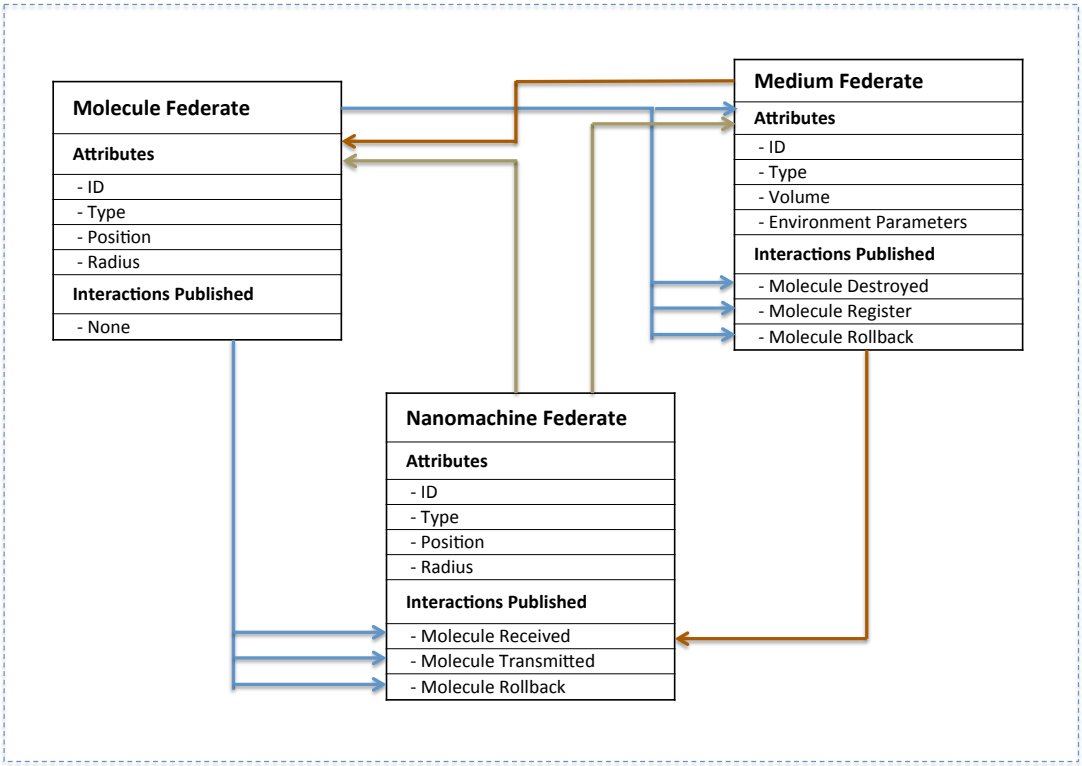


Figure 3.10. Federates.

(a) *Published Object Class Attributes*

- *ID*: A globally unique identifier that is automatically generated by the federate once a new molecule is transmitted.
- *Type*: The type of the molecule, for representing its behaviour in the medium. Different types are used for modelling different molecules.

- *Position*: The position of the molecule in 3-D space.
- *Radius*: The radius of the molecule.

(b) *Subscribed Object Class Attributes and Interactions*

The Molecule Federate subscribes to the *ID*, *Type*, *Volume*, and *Environment Parameters* attributes and the *Molecule Destroyed*, *Molecule Register*, and *Molecule Rollback* interactions published by the Medium Federate, and *Molecule Received*, *Molecule Transmitted*, and the *Molecule Rollback* interactions published by the Nanomachine Federate.

Require: Federation is created
join and synchronize with federation
for *timestep* = 0 to *simulationtime* **do**
  if *timestep* is intermediate **then**
    for all *molecule* in *moleculerollbackinteraction* **do**
      rollback (*molecule*)
    end for
  else
    create and register new molecules
    for all *molecule* in *existingmolecules* **do**
      update location (*molecule*)
    end for
    for all *molecule* in *moleculereceivedinteraction* **do**
      delete (*molecule*)
    end for
    for all *molecule* in *moleculedestroyedinteraction* **do**
      delete (*molecule*)
    end for
  end if
end for
resign from federation

Figure 3.11. Algorithm for MoleculeFederate.

(ii) *Nanomachine Federate:* The Nanomachine Federate models the nanomachines in

the simulation. It is responsible for implementation of molecule transmission and reception mechanisms. It is possible to define different types of transmitter and receiver nanomachines with different behaviours. Different implementations of a transmitter nanomachine can release molecules instantaneously or sequentially, from a single point on its surface, or from multiple points. An implementation of a receiver nanomachine can receive molecules all over its surface, or another implementation can receive only via receptors distributed on its surface. Our current implementation for transmitter releases molecules from a single point on its surface. The molecules are released at the beginning of symbol duration instantaneously. The receiver nanomachine is implemented to receive molecules as soon as they collide with the nanomachine surface, or if receptors are defined, the molecules are received only when they collide with one of the receptor surfaces [56]. The published and subscribed attributes and interactions defined for the nanomachine object in FOM are listed below. The published attributes and interactions are communicated to other federates which subscribe to these attributes or interactions. This relation is shown in Figure 3.10 and the pseudocode for the Nanomachine Federate is given in Algorithm 3.12.

(a) *Published Object Class Attributes*

- *ID*: The identifier of the nanomachine. In our implementation, the unique identifier of the nanomachine is read from a configuration file, which also contains other parameters of the nanomachine.
- *Type*: The type of the nanomachine. Currently two types are supported, transmitter and receiver.
- *Position*: The position of the nanomachine in 3-D space.
- *Radius*: The radius of the nanomachine.

(b) *Published Interactions*

- *Molecule Received*: This interaction is sent when the molecule is received by a nanomachine.
- *Molecule Rollback*: This interaction is sent by a nanomachine if a molecule tries to move to a position that is not available due to the transmitter nanomachine. In that case, the molecule performs a rollback in an intermediate time step.

- *Molecule Transmitted*: This interaction is sent by a nanomachine to transmit molecules.

(c) *Subscribed Object Class Attributes and Interactions*

The Nanomachine Federate subscribes to the *ID*, *Type*, *Volume*, and *Environment Parameters* attributes published by the Medium Federate, and the *ID*, *Type*, *Position*, and *Radius* attributes published by the Molecule Federate.

Require: Federation is created
join and synchronize with federation
for *timestep* = 0 to *simulationtime* **do**
  if *timestep* is intermediate **then**
    for all *molecule* collided with a transmitter nanomachine **do**
      send interaction (*moleculerollbackinteraction*)
    end for
    for all *molecule* received by a receiver nanomachine **do**
      send interaction (*moleculereceivedinteraction*)
    end for
  else
    if *nanomachine* is transmitter **and** *timestep* is start of *symbolduration* **then**
      send interaction (*moleculetrasnmittedinteraction*)
    else if *nanomachine* is receiver **and** *timestep* is end of *symbolduration* **then**
      save received molecule statistics
    end if
  end if
end for
resign from federation

Figure 3.12. Algorithm for NanomachineFederate.

(iii) *Medium Federate:* The Medium Federate abstracts the medium slices. It is possible to define medium slices to construct a larger 3-D space for simulation. It also performs collision handling for molecules. Our implementation includes a basic collision management, in which the molecule movements are evaluated in random

order by the Medium Federate. If the target location is occupied by another molecule, a rollback is performed at an intermediate time step. It is also possible to extend this implementation to implement more complex collision management by considering underlying physical and chemical laws during a collision process. The simulation scalability can be achieved by assigning different medium slices to different medium federates. The published and subscribed attributes and interactions defined for the medium object in FOM are listed below. The published attributes and interactions are communicated to other federates which subscribe to these attributes or interactions. This relation is shown in Figure 3.10 and the pseudo-code for the Medium Federate is given in Algorithm 3.13.

(a) *Published Object Class Attributes*

- *ID*: The identifier of the medium slice. In our implementation, the medium slices are created based on configuration files that are uniquely identified by medium slice IDs, and the configuration files contain the configuration parameters of the medium slices.
- *Type*: The type of the medium slice.
- *Volume*: The parameter indicating the 3-D volume of the slice.
- *Environmental Parameters*: The parameters specific to a medium slice.

(b) *Published Interactions*

- *Molecule Destroyed*: This interaction is sent by the medium once the molecule diffuses out of simulation boundaries.
- *Molecule Register*: This interaction is sent by the medium once the molecule goes inside a medium slice. Once registered to the medium, the molecule starts its movements according to the medium parameters.
- *Molecule Rollback*: This interaction is sent by the medium if a molecule tries to move to a position that is not available due to a collision. In that case, molecule performs a rollback in an intermediate time step.

(c) *Subscribed Object Class Attributes and Interactions*

The Medium Federate subscribes to the *ID*, *Type*, *Position*, and *Radius* attributes published by the Molecule Federate and the *Molecule Received* interaction published by the Nanomachine Federate.

Require: Federation is created

1: join and synchronize with federation
2: **for** *timestep* = 0 to *simulationtime* **do**
3: **if** *timestep* is intermediate **then**
4: **for all** *molecule* collided with another molecule **do**
5: send interaction (*moleculerollbackinteraction*)
6: **end for**
7: **for all** *molecule* out of simulation boundaries **do**
8: send interaction (*moleculedestroyedinteraction*)
9: **end for**
10: **for all** *molecule* registered for the medium **do**
11: send interaction (*moleculeregisteredinteraction*)
12: **end for**
13: **end if**
14: **end for**
15: resign from federation

Figure 3.13. Algorithm for MediumFederate.

3.3.3. Simulation Experiments

To demonstrate the simulation tool, the scenario shown in Figure 3.14 is used. In this scenario, there are eight transmitter nano-machines and eight receiver nano-machines distributed in a 3-D space. The simulation parameters are listed in Table 3.3. The molecules are transmitted by the transmitter nano-machines at the start of the symbol durations, each emitting one hundred molecules to represent binary *one* and zero to represent binary *zero*. The input sequences are randomly generated, and the same sequence is used for all transmitter nano-machines. Once transmitted, the molecules diffuse through medium, and if they hit a receiver nano-machine, they are received by that nano-machine. The molecules diffuse from one medium slice to another and if a molecule diffuses out of simulation boundaries, it is removed from the simulation.

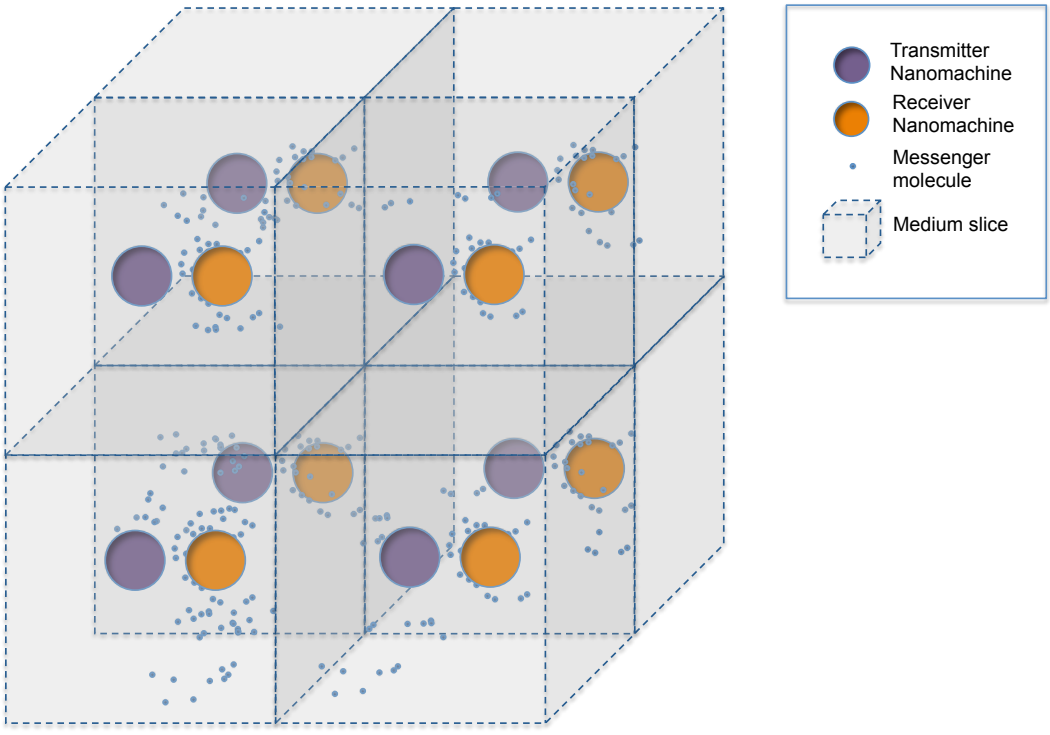


Figure 3.14. Simulation Scenario for Performance Evaluation of Distributed Architecture.

For the performance evaluation of the distributed simulation, we define the speedup,

Table 3.3. Simulation Parameters.

Parameter	**Value**
Temperature	310 ^oK
Viscosity of the fluid	0.001 kg/s.m
Medium dimensions (x/y/z)	200/200/200 μm
Radius of the Transmitters	10 μm
Radius of the Receivers	10 μm
Radius of the Molecule	2.86 nm
Distance between the transmitter and the receiver	1 μm
Symbol duration	30 ms

S, as

$$S = \frac{T_s}{T_m} \tag{3.14}$$

where T_s is the execution time with a single federate and T_m is the execution time with multiple federates. The simulation tool is deployed on a cluster of Linux nodes communicating over a Local Area Network. The analysis is done for scalability of Medium Federates, Nanomachine Federates, and Molecule Federates, with single, two, four, and eight federates for each. A separate Linux node is used to run each Federate. If there is a single Medium Federate, it manages all the simulation space. When there are two federates, the simulation space is sliced into two and each federate manages one slice. A similar logic is used for four and eight Medium Federates. For Molecule Federates, each federate manages movements of a percentage of molecules in the medium. Similarly, each Nanomachine Federate manages the nano-machine objects assigned to the federate. Each simulation is repeated ten times to calculate the average speedup value.

Firstly, in order to verify the smooth operation of the simulator, we use one of the basic analytical models for random walk. Consider a spherical shell source of radius b

between a spherical absorber of radius a and a spherical shell absorber of radius c as shown in Figure 3.15. The probability that a particle released at $r = b$ will be adsorbed at $r = a$ is given by

$$P_{hit} = \frac{a(c-b)}{b(c-a)}. \tag{3.15}$$

At the limit where $c \to \infty$, this probability is a/b [42].

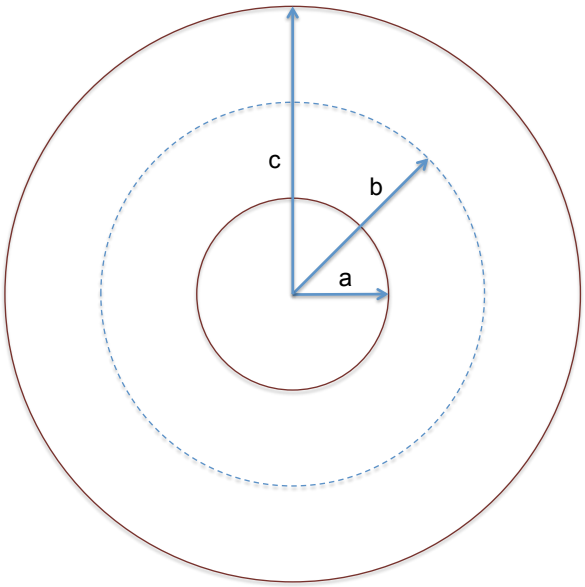


Figure 3.15. Probability of Capture.

We use the same scenario, and approximate the model by selecting c as $100 \cdot a$. Figure 3.16 shows the probability of hit versus the distance between the source and the destination for different simulated time values, and the results of the analytical model. As simulated time increases, the probability of hit approaches to analytical value, and since the mean time of arrival for greater distances is higher, the effect of simulated time to the probability of hit is also higher. The results show that the functional implementation of transmission, random walk and reception is in parallel to the given analytical model.

In addition, to verify the smooth operation of the simulation under distributed environment, we analyse the sent and the received molecules for the simulation executions with an increasing number of medium federate nodes. For an input sequence of

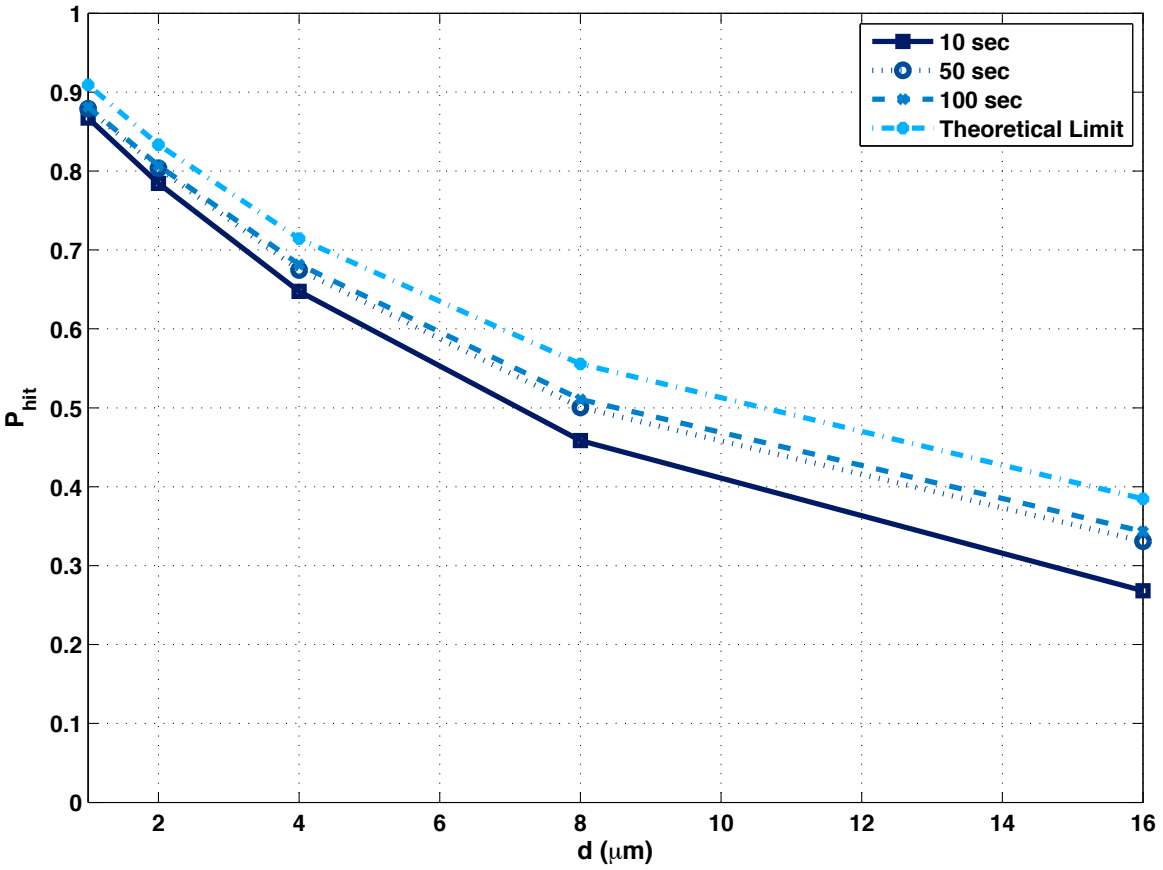


Figure 3.16. Probability of Capture Results.

'1101101011', Figure 3.17 shows the number of transmitted and received molecules for each time slot. For each time slot, the number of transmitted and received molecules within the symbol duration is plotted. A similar pattern is observed for an increasing number of nodes. The variation is due to the random nature of Brownian motion, and the pattern shows that similar output is generated when the level of parallelism is increased.

3.3.3.1. Collision and Collision Free Implementation. We run the simulation scenario for a collision free medium, where the molecules can coexist at the same position, and also with the collision management mechanism that we implemented. Our collision management implementation is a basic one in which the molecule movements are evaluated in random order, and if the target location is occupied by another molecule, the movement is not performed. This implementation is for demonstrating the effect of a collision implementation over speedup. The results are depicted in Figure 3.18. For

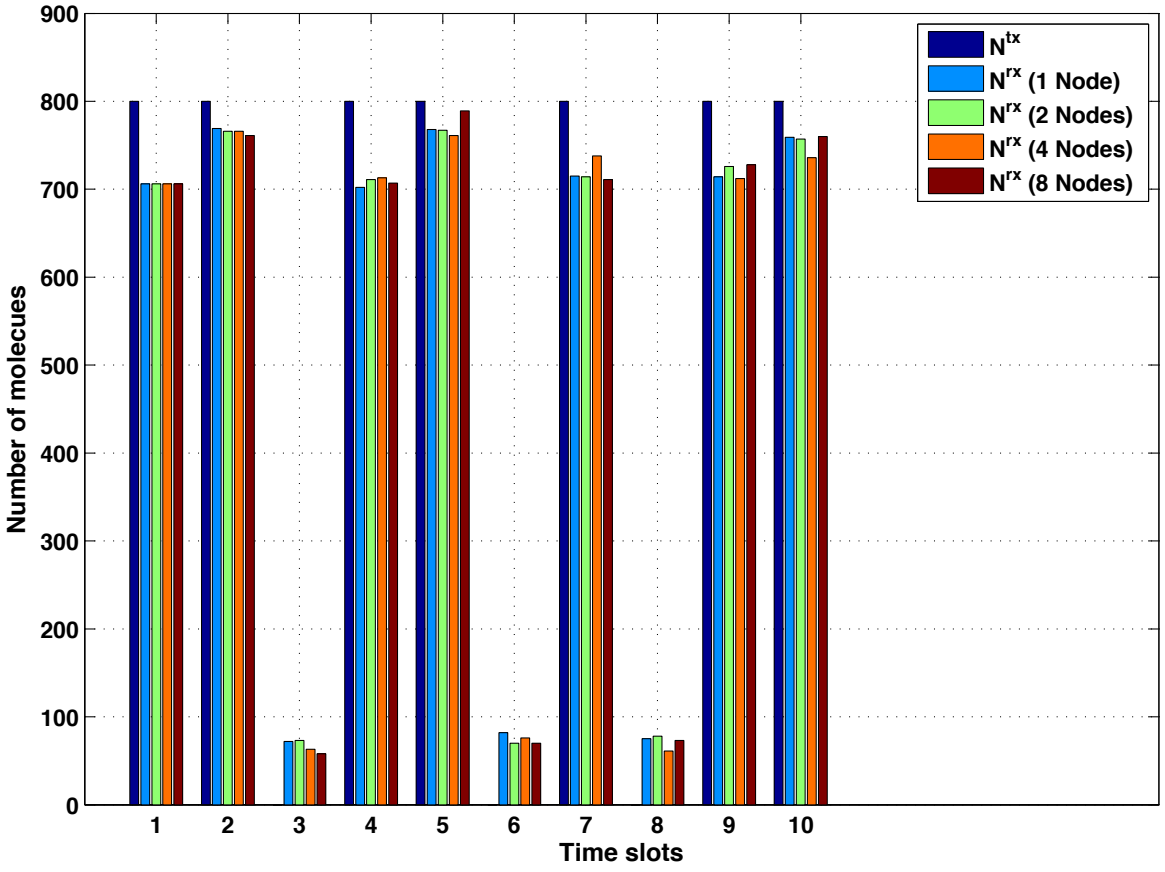


Figure 3.17. Number of Transmitted and Received Molecules.

the collision free scenario, increasing the number of medium federates does not result in a speedup, while with the collision implementation, a speedup is achieved. This result shows that for the collision free scenario, the computational bottleneck is not in collision evaluation, and the gain of multiple nodes is lost on communication overhead among nodes. This shows that for such scenarios, to benefit from the distributed architecture, the load on the Molecule Federate and the Nanomachine Federate needs to be distributed over multiple nodes. The results for such scenarios are included in the following sections. For collision implementation, the computational load is on collision management, since each medium slice should evaluate the movements and run the collision algorithm. When this intensive computational work is distributed over multiple nodes, execution time decreases. As seen from Figure 3.18, the efficiency of the parallelism decreases after four nodes, since the communicational overhead becomes dominant.

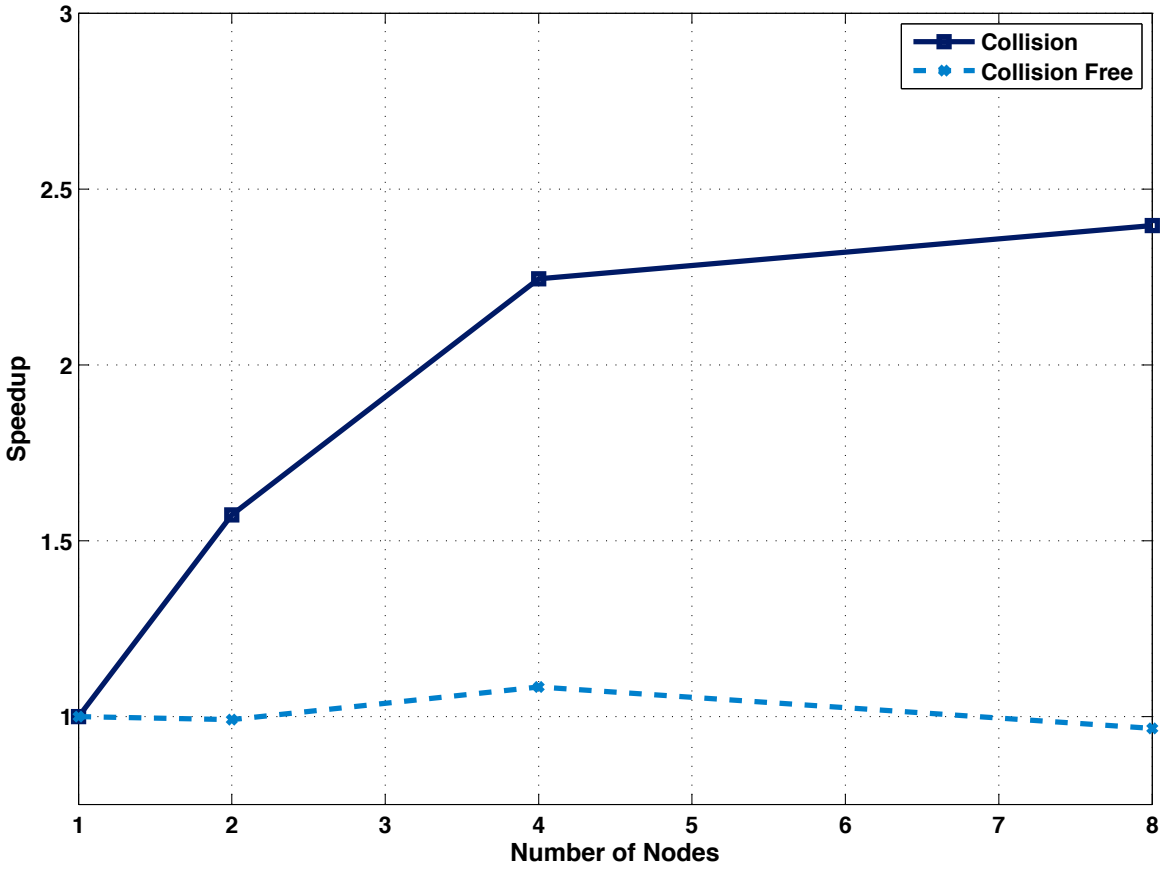


Figure 3.18. Collision and Collision Free.

3.3.3.2. Effect of Simulation Duration. To understand the effect of simulation duration, we execute the simulation scenario with collisions, for different simulation durations. The results in Figure 3.19 show that, as the simulation duration increases, the speedup also increases. This can be explained by the increase in computational load to evaluate collisions due to the molecules accumulating in the medium. It is also observed that when simulation time is increased, in parallel to the increase in speedup, the effect of communication overhead is also experienced earlier. Hence, there is nearly no gain for eight nodes compared to four nodes for a longer simulation duration.

3.3.3.3. Effect of Collision Algorithm Complexity. The results elaborated in the previous sections show that the distribution of the simulation to multiple nodes results in higher speedups when a computationally intensive collision handling algorithm is implemented. To analyse this effect better, we implement dummy algorithms, which require 1 ms, 2 ms, 4 ms, 8 ms and 16 ms of computational time to execute per molecule

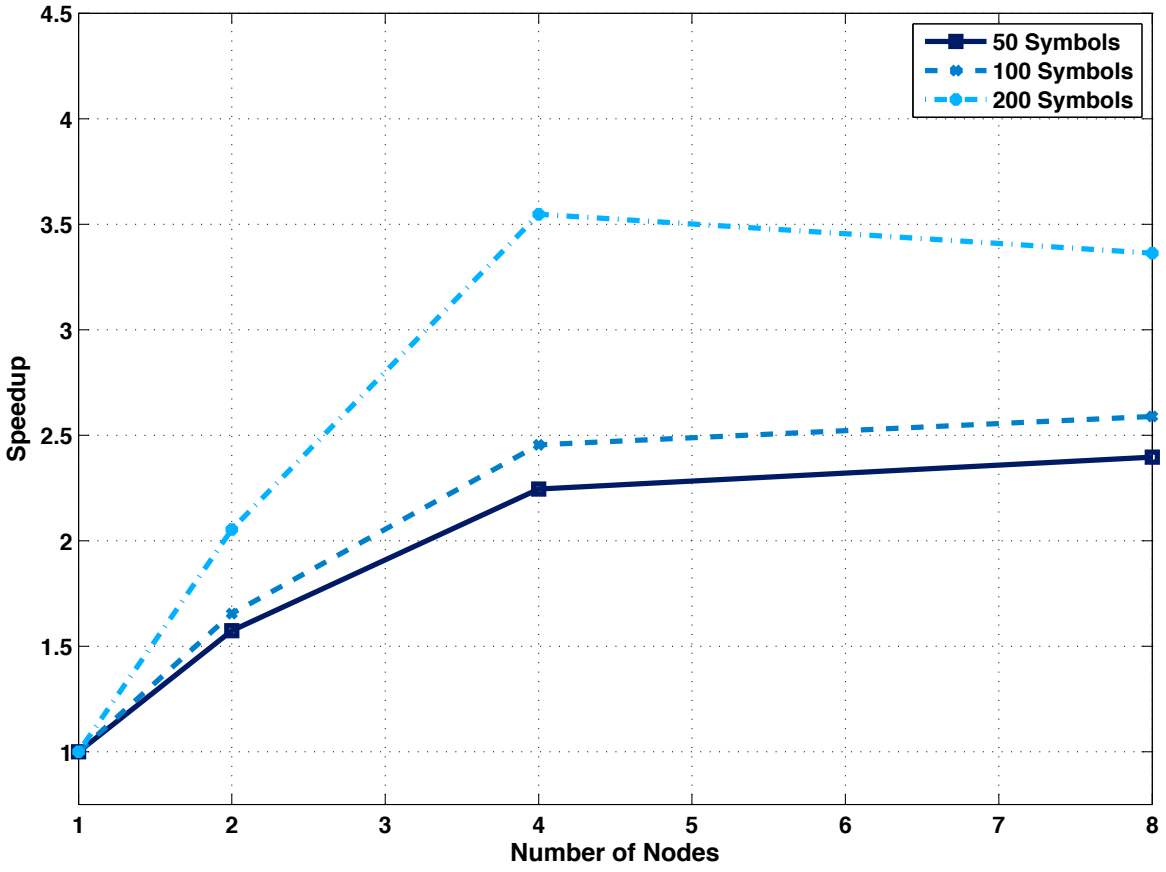


Figure 3.19. Effect of Simulation Duration.

movement. The results in Figure 3.20 show that, as the computational time for the medium federate increases, the system performance increases up to 8 ms. After that point, it does not get better towards the theoretical linear speedup limit due to the communication overhead. In the scenarios where a high computational load is required for collision management, the system achieves close to linear speedup.

3.3.3.4. Nanomachine Federate Scalability. For the cases where the bottleneck is not in the collision management, different parallelism options need to be considered. This can be achieved by using multiple Nanomachine Federates, to manage a set of nanomachines. The scalability of the simulation will be dependent on the complexity of the algorithms used for the transmission and the reception processes.

We start with the analysis of the effect of receptor implementation. Figure 3.21 shows the effect of the distributed execution for basic reception mechanism, in which

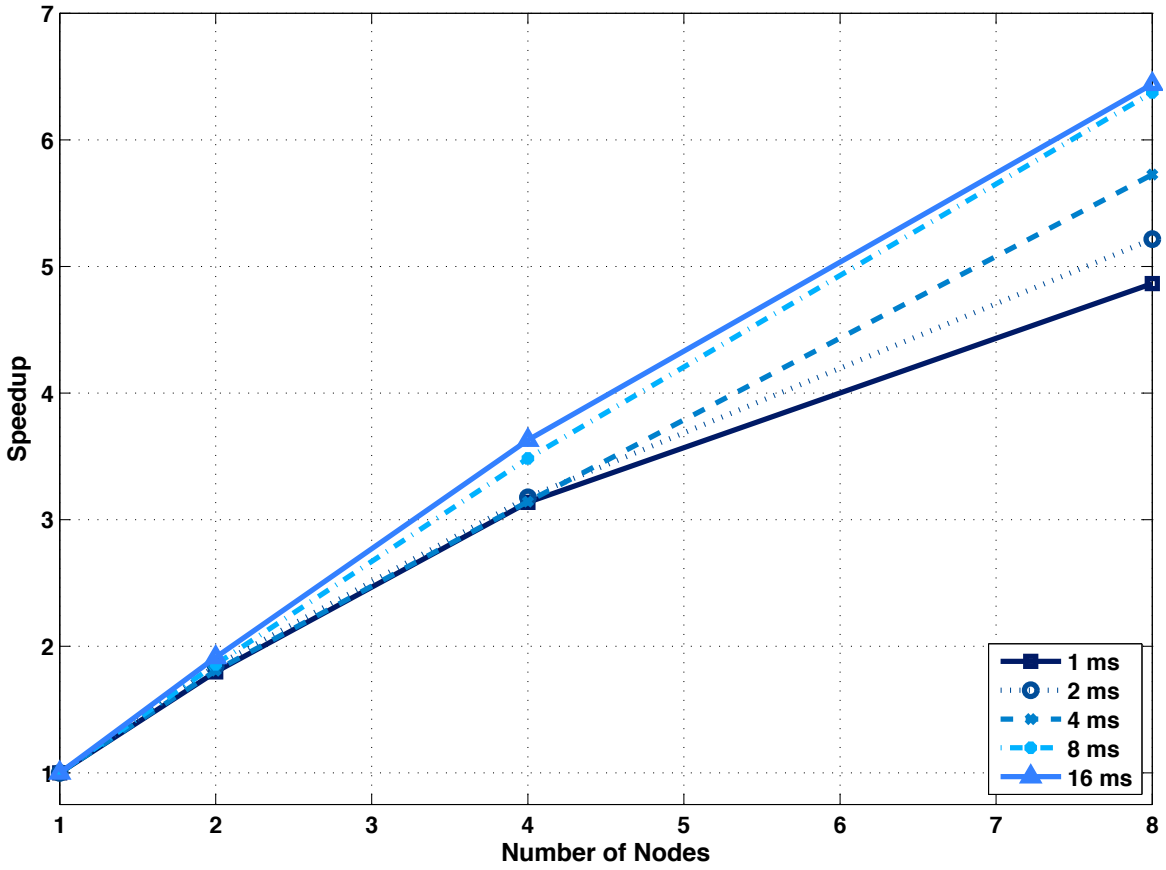


Figure 3.20. Effect of Algorithm Complexity.

a molecule is received by the nano-machine when it collides with the receiver nano-machine surface, as well as the effect of multiple receptors randomly distributed over the receiver nano-machine surface. In the nature, the number of cell surface receptors for signalling molecules can vary from 500 to more than 100 000 per cell for a specific signalling molecule [5]. Figure 3.21 shows the results for 1 000, 10 000, and 100 000 receptors. when a simple reception in which whole surface is used for reception is implemented, adding more Nanomachine Federates does not decrease the execution time since the additional processing power can not be utilized due to the networking overhead. The networking overhead becomes more dominant as the number of nodes increases. As the number of receptors increases, the computational power required for evaluating reception increases, hence the speedup also increases.

Figure 3.22 shows the results for dummy algorithms which take $10ms$, $20ms$, $40ms$, $80ms$, and $160ms$ for reception processes. When more complex algorithms

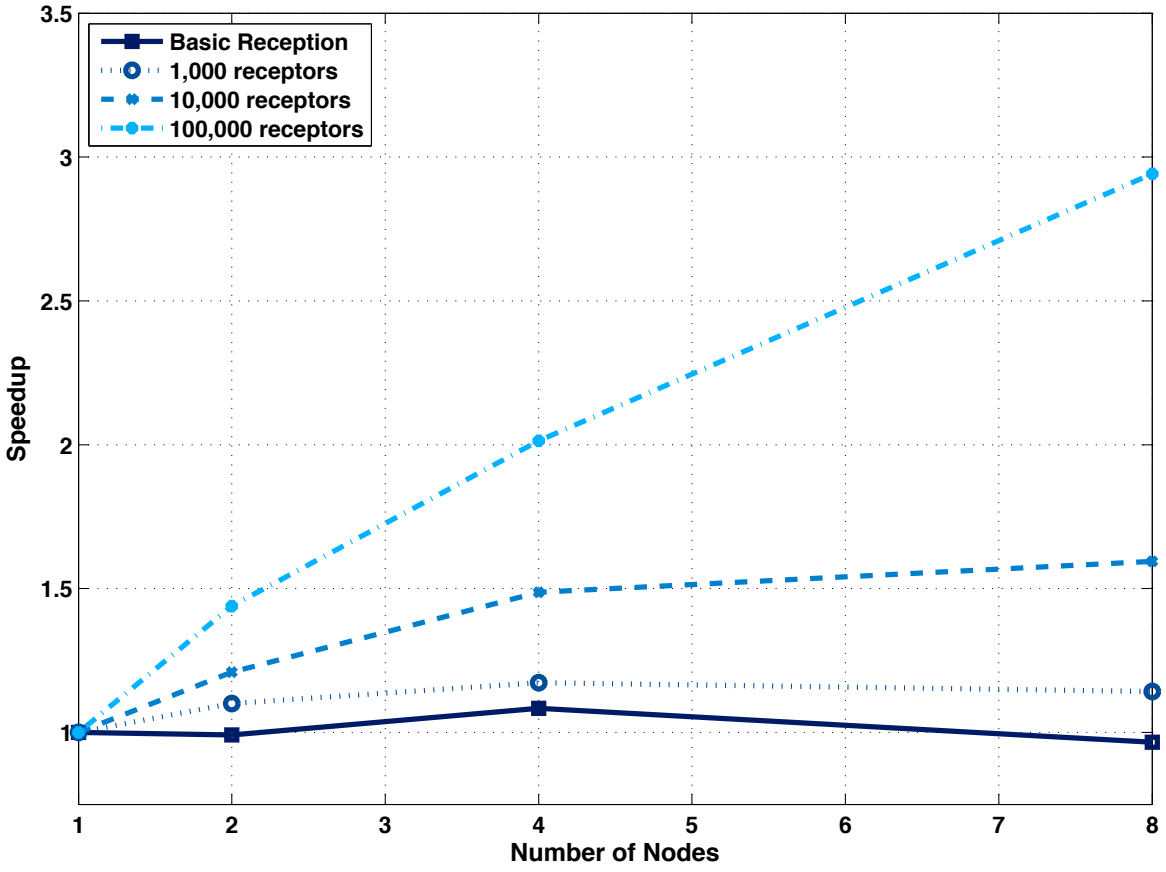


Figure 3.21. Receptor Implementation Scalability.

are implemented, the system speedup increases as the complexity of the algorithm increases. This shows that, if complex transmission or reception models need to be simulated, the scalability of the system should be achieved using multiple Nanomachine Federates.

3.3.3.5. Molecule Federate Scalability. To analyze the simulation scenarios where the molecule movements model is complicated, we compare Brownian motion and more complex dummy algorithms, which require specific processing time. As shown in Figure 3.23, for Brownian motion, the network overhead is dominant and no gain is achieved in terms of execution time. When more complex algorithms are implemented, which take 1 ms, 2 ms, 4 ms, 8 ms, and 16 ms for molecule movement evaluation, better system speedup is observed. The results show that scalability of Molecule Federates need to be used for simulation scenarios where computationally intensive movement models need to be analysed.

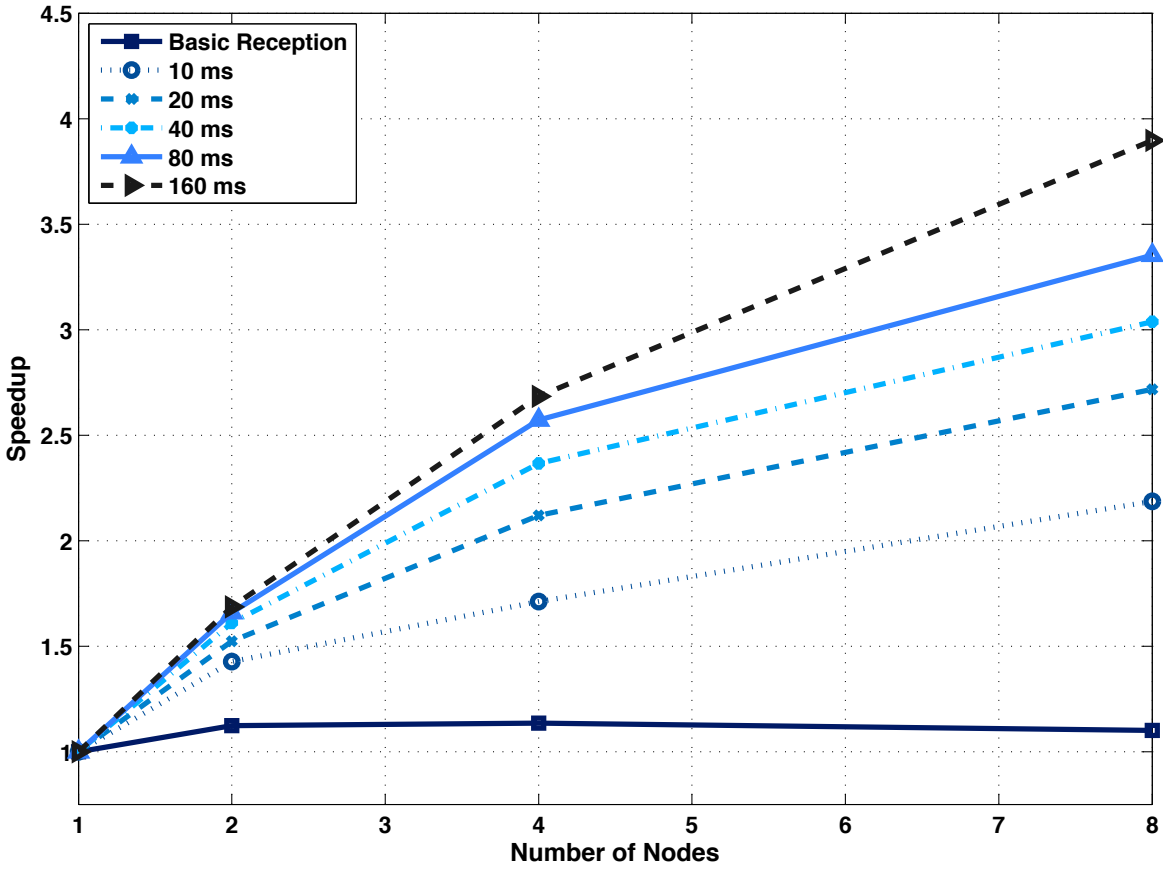


Figure 3.22. Generic Nanomachine Federate Scalability.

3.4. Chapter Summary

Simulation is an important tool to analyse different topologies for molecular communication via diffusion, either to verify the analytical analysis or for the scenarios where analytical formulations are not available. The design and the execution of the simulations are important steps in research activities. The researchers should decide on the abstraction level carefully to optimize accuracy and execution time. Since simulation execution time is an important bottleneck for research projects, we introduce two models to address the issue. A dual-zone based simulation model can be used to decrease simulation execution time, while keeping the accuracy at an acceptable level. The results show that, for scenarios where the number of molecules in the environment is high, the proposed model achieves significant speed up values. We also introduce an HLA-based simulation architecture for molecular communication. The architecture enables defining inter-operable and reusable simulation components and makes the ex-

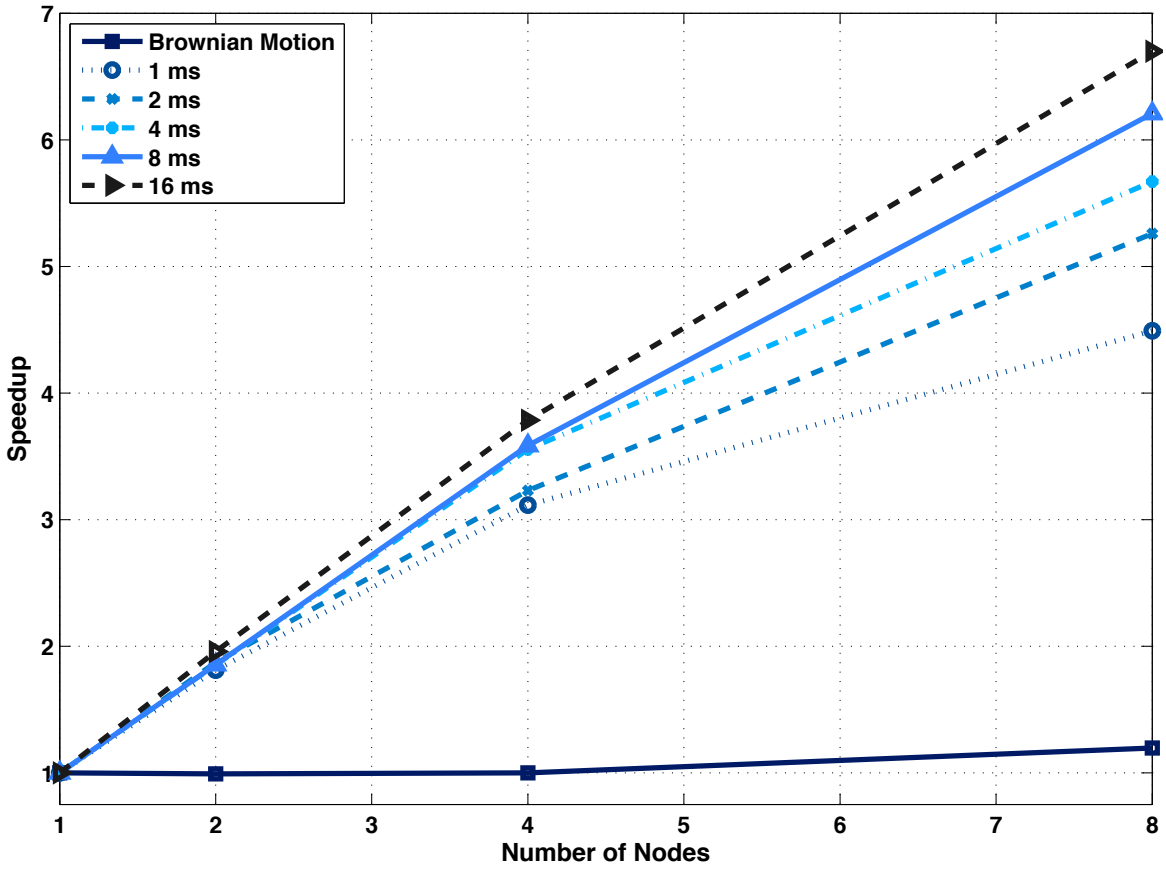


Figure 3.23. Molecule Federate Scalability.

ecution of large scale hybrid scenarios possible. For computationally intensive models, the benefits of distributed architecture is fully utilized.

4. RECEPTION PROCESS FOR AN ABSORBING RECEIVER

In the nature, one of the most useful example of diffusion process is a diffusing molecule around a target, which is either absorbed by the target or vanishes out. Using only concentration function without considering absorption process assumes that the receiver does not affect the system. We need to consider the absorption process to better represent the real reception phenomena. This chapter aims to analyse the reception process for an absorbing spherical receiver.

4.1. Communication Model

Figure 4.1 shows the communication model we use in this chapter. Messenger molecules are used as the information carriers between a point source and an absorbing spherical receiver. The point source is located at a distance r_0 from the center of the receiver. The point source and the spherical receiver both reside in a fluid propagation medium. It is assumed that the medium is unconfined, thus extending to infinity in all directions. After the information is modulated onto some physical property of the molecules, the molecules are released to the medium where they diffuse according to Brownian motion and arrive at the receiver. The spherical receiver with radius r_r, uses all its surface to absorb the molecules. If a molecule collides with the surface of the receiver it is absorbed by it.

4.2. Absorption Rate of a Spherical Receiver

The microscopic theory of diffusion roots from the assumption that a substance will move down its concentration gradient. The derivative of the flux with respect to time results in Fick's Second Law in a 3-D environment, given by

$$\frac{\partial p(r,t|r_0)}{\partial t} = D\nabla^2 p(r,t|r_0) \tag{4.1}$$

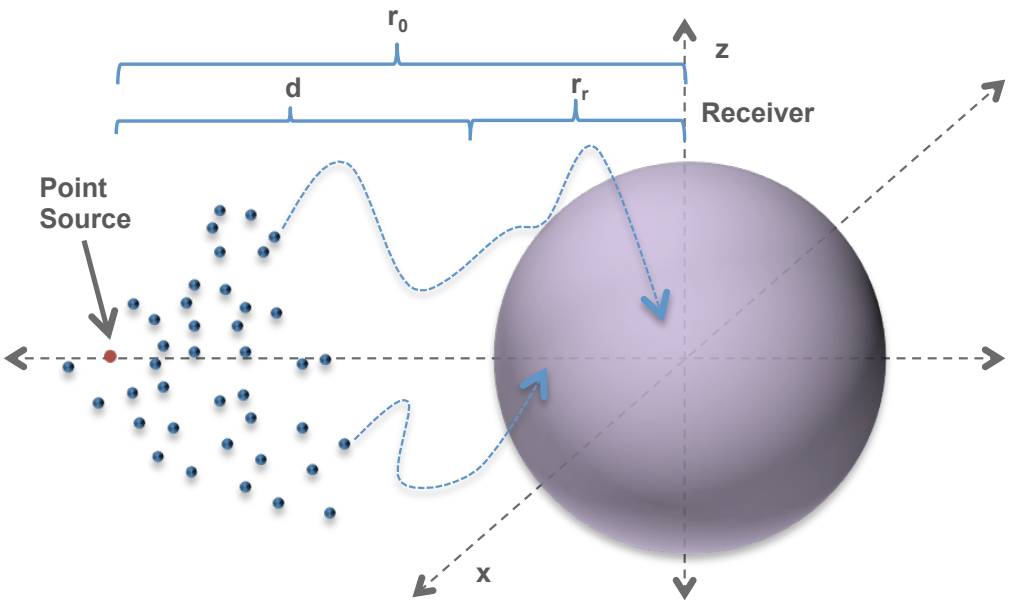


Figure 4.1. Communication model for an absorbing spherical receiver.

where ∇^2, $p(r,t|r_0)$, and D are the Laplacian operator, the molecule distribution function at time t and distance r given the initial distance r_0, and the diffusion constant, respectively. The value of D depends on the temperature, viscosity of the fluid, and the Stokes' radius of the molecule [44].

The fraction of molecules hitting to a spherical receiver located at $(0,0,0)$ to the total number of released molecules can be derived by solving the Fick's diffusion equation with the initial and the boundary conditions obeying the problem and describing the absorbing process following the methodologies in [40, 43, 57].

The initial condition is defined as

$$p(r, t \to 0|r_0) = \frac{1}{4\pi r_0^2}\delta(r - r_0), \tag{4.2}$$

and the first boundary condition is

$$\lim_{r\to\infty} p(r,t|r_0) = 0, \tag{4.3}$$

which reflects the assumption that the distribution of the molecules vanishes at distances far greater than r_0. The second boundary condition is

$$D\frac{\partial p(r,t|r_0)}{\partial r} = w\, p(r,t|r_0) \text{ , for } r = r_r \tag{4.4}$$

where r_r and w denote the radius of the receiver and the rate of reaction. Reaction rate with the receiver boundary is controlled by w. $w = 0$ means a nonreactive surface while w approaching to infinity corresponds to the boundary in which every collision leads to an absorption.

We observe that Fick's Second Law given in (4.1) becomes

$$\frac{\partial(r \cdot p(r,t|r_0))}{\partial t} = D\frac{\partial^2 r \cdot p(r,t|r_0)}{\partial r^2} \tag{4.5}$$

when we move to the spherical coordinated and drop the terms with θ and ϕ from the Laplacian operator, since $p(r,t|r_0)$ is spherically symmetric and solely depends on r.

Next step is to partition $p(r,t|r_0)$ into two equations $u(r,t|r_0)$ and $v(r,t|r_0)$, which both obey the diffusion equation (4.5) and together obey the boundary conditions given in (4.3) and (4.4). Since they obey the boundary conditions $u(r,t|r_0)$ must satisfy

$$\frac{\partial(r \cdot u(r,t|r_0))}{\partial t} = D\frac{\partial^2 r \cdot u(r,t|r_0)}{\partial r^2} \tag{4.6}$$

$$r \cdot u(r, t \to 0|r_0) = \frac{1}{4\pi r_0}\delta(r - r_0), \tag{4.7}$$

Using Fourier transform, one can obtain

$$r \cdot u(r,t|r_0) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \mathcal{U}(k,t|r_0)e^{ikr}dk. \tag{4.8}$$

When we plug (4.8) into (4.5) we obtain

$$\mathcal{U}(k,t|r_0) = K_u \cdot \exp[-Dtk^2] \tag{4.9}$$

where K_u is the time dependent coefficient. It is determined from the initial condition (4.7) as

$$K_u = \frac{1}{4\pi r_0} e^{ikr_0}, \tag{4.10}$$

which results in the final Fourier expression as

$$r \cdot u(r,t|r_0) = \frac{1}{8\pi^2 r_0} \int_{-\infty}^{+\infty} \exp[-Dtk^2] e^{ik(r-r_0)} dk, \tag{4.11}$$

that yields the following expression after integration

$$r \cdot u(r,t|r_0) = \frac{1}{4\pi r_0} \frac{1}{\sqrt{4\pi Dt}} \exp\left[-\frac{(r-r_0)^2}{4Dt}\right]. \tag{4.12}$$

Then, we handle the second part, $v(r,t|r_0)$, which must satisfy

$$\frac{\partial(r \cdot v(r,t|r_0))}{\partial t} = D \frac{\partial^2 r \cdot v(r,t|r_0)}{\partial r^2} \tag{4.13}$$

$$r \cdot v(r, t \to 0|r_0) = 0. \tag{4.14}$$

Through Laplace transform we obtain

$$\frac{s}{D}(r \cdot \mathcal{V}(r,s|r_0)) = \frac{\partial^2 (r \cdot \mathcal{V}(r,s|r_0))}{\partial r^2} \tag{4.15}$$

where $\mathcal{V}(r,s|r_0)$ is the Laplace transform of $v(r,t|r_0)$. Applying the boundary condition

(4.3), we obtain

$$r \cdot \mathcal{V}(r, s|r_0) = K_v \cdot \exp\left[-\sqrt{\frac{s}{D}}r\right], \tag{4.16}$$

where K_v is a constant that should satisfy the second boundary condition (4.4). We finally plug in the Laplace transform of $u(r, t|r_0)$ to obtain

$$r \cdot \mathcal{P}(r, s|r_0) = r \cdot \mathcal{U}(r, s|r_0) + r \cdot \mathcal{V}(r, s|r_0) \tag{4.17}$$

$$= \frac{1}{4\pi r_0}\frac{1}{\sqrt{4Ds}} \exp\left[-\sqrt{\frac{s}{D}}\,|r - r_0|\right] + K_v \exp\left[-\sqrt{\frac{s}{D}}r\right]. \tag{4.18}$$

With the Laplace transform of the boundary condition (4.4) we get

$$\left.\frac{\partial(r \cdot \mathcal{P}(r, s|r_0))}{\partial t}\right|_{r=r_0} = \frac{wr_r + D}{Dr_r} r_r \cdot \mathcal{P}(r, s|r_0). \tag{4.19}$$

Using (4.17) and (4.19) we obtain

$$K_v = \frac{\sqrt{\frac{s}{D}} - \frac{wr_r+D}{Dr_r}}{\sqrt{\frac{s}{D}} + \frac{wr_r+D}{Dr_r}} \frac{1}{4\pi r_0}\frac{1}{\sqrt{4Ds}} \exp\left[-\sqrt{\frac{s}{D}}(r_0 - 2r_r)\right], \tag{4.20}$$

from which we can calculate $\mathcal{P}(r, s|r_0)$. The inverse Laplace transform of $\mathcal{P}(r, s|r_0)$ yields to

$$\begin{aligned} p(r, t|r_0) = {} & \frac{1}{4\pi r r_0}\frac{1}{\sqrt{4\pi Dt}} \exp\left[-\frac{(r - r_0)^2}{4Dt}\right] + \exp\left[-\frac{(r + r_0 - 2r_r)^2}{4Dt}\right] \\ & - \frac{1}{4\pi r r_0}\frac{wr_r + D}{Dr_r} \exp\left[\left(\frac{wr_r + D}{Dr_r}\right)^2 Dt + \frac{wr_r + D}{Dr_r}(r + r_0 - 2r_r)\right] \\ & \times \operatorname{erfc}\left[\frac{wr_r + D}{Dr_r}\sqrt{Dt} + \frac{r + r_0 - 2r_r}{4Dt}\right] \end{aligned} \tag{4.21}$$

For an absorbing receiver, we should consider the case where $w \to \infty$, and (4.21)

becomes

$$p(r,t|r_0) = \frac{1}{4\pi r r_0}\frac{1}{\sqrt{4\pi Dt}} \times \left(\exp\left[-\frac{(r-r_0)^2}{4Dt}\right] - \exp\left[-\frac{(r+r_0-2r_r)^2}{4Dt}\right]\right). \quad (4.22)$$

Following the molecule distribution $p(r,t|r_0)$, hitting rate of the molecules to the receiver at time t can be obtained as,

$$f_{\text{hit}}(t) = 4\pi r_r^2 p(r_r,t|r_0) \quad (4.23)$$

$$= \frac{r_r}{r_0}\frac{1}{\sqrt{4\pi Dt}}\frac{r_0-r_r}{t}\exp\left[-\frac{(r_0-r_r)^2}{4Dt}\right]. \quad (4.24)$$

Furthermore, by integrating $f_{\text{hit}}(t)$ with respect to time, we can obtain $F_{\text{hit}}(t)$, which is the fraction of molecules absorbed by the receiver until time t:

$$F_{\text{hit}}(t) = \frac{r_r}{r_0}\text{erfc}\left[\frac{r_0-r_r}{\sqrt{4Dt}}\right]. \quad (4.25)$$

Using (4.25), we define the fraction of molecules received between t_1 and t_2 as

$$F_{\text{hit}}(t_1,t_2) = F_{\text{hit}}(t_2) - F_{\text{hit}}(t_1). \quad (4.26)$$

To analyse the formulations, we use a simulation scenario where the radius of the receiver is $10\mu m$, the diffusion coefficient is $79.4\mu m^2/s$, and time step is 0.0001 s. The simulation is repeated for 100 000 molecules to calculate the hitting rate via simulation.

Figure 4.2 shows the simulation results and analytical formulation for $d = 1\mu m$, which means the point source is located at a distance of $r_0 = 11\mu m$ from the centre of the receiver. The results are depicted as a histogram with granularity of 10^{-3}. For $d = 1\mu m$, the peak value is observed right after the release of the molecule. The simulation results and the analytical results overlap. The only small deviation is where the peak is observed at the beginning of the simulation. The reason of this is due to

the relatively large time step of 0.0001 s. Some of the molecules are not captured due to large time step as described in Section 3.3.

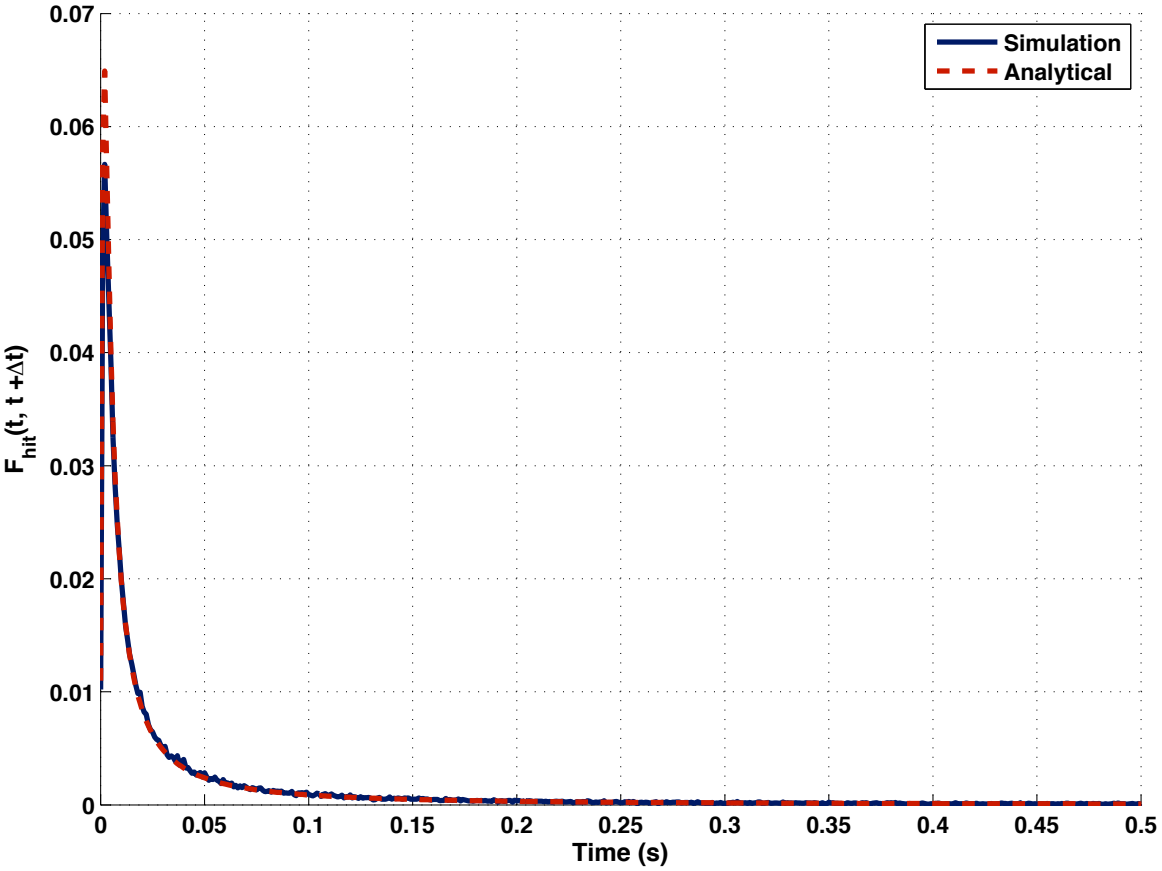


Figure 4.2. Analytical and simulation results for the fraction of molecules absorbed by the receiver versus time for $d = 1\mu m$.

In Figure 4.3, the simulation results and analytical formulation are depicted for $d = 2\mu m$, which means the point source is located at a distance of $r_0 = 12\mu m$ from the centre of the receiver. For $d = 2\mu m$, the peak value is observed later than the peak value of $d = 1\mu m$ as expected. Since the distance is higher, the hitting rate is smaller, and molecules scatter around more. Due to this fact, the variation of the simulation results becomes a bit higher. Again a deviation at peak value is observed due to the relatively large time step of 0.0001 s. Some of the molecules are not captured due to large time step as described in Section 3.3.

Figure 4.4 shows the simulation results and analytical formulation for $d = 4\mu m$ and $d = 8\mu m$, which means the point source is located at a distance of $r_0 = 14\mu m$

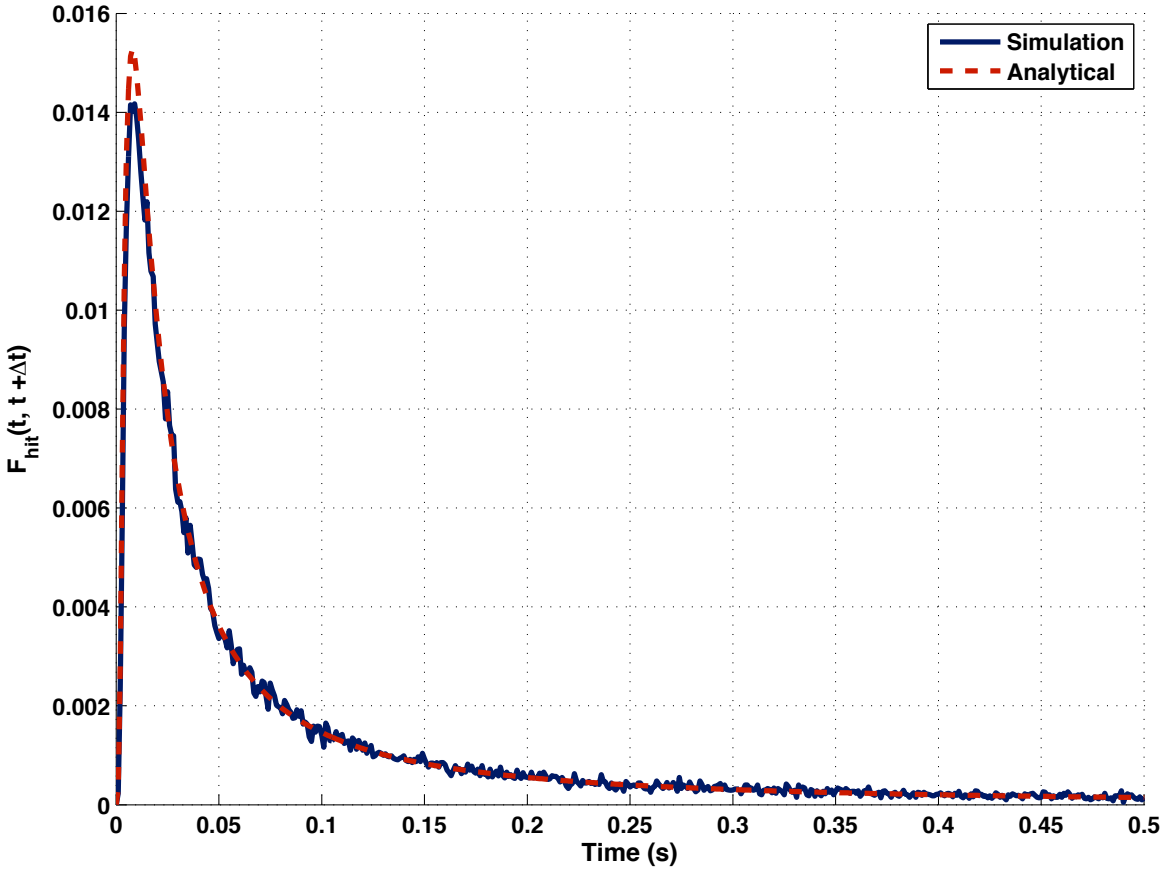


Figure 4.3. Analytical and simulation results for the fraction of molecules absorbed by the receiver versus time for $d = 2\mu m$.

and $r_0 = 18\mu m$ from the centre of the receiver, respectively. As the distance increases, the first hitting time and the peak value increases. The molecules also scatter to the environment more since the hitting rate also decreases. This is why the variation of the simulation results increases as well. For higher distances, the effect of time step becomes smaller since less molecules are able to reach around the receiver. This is why the fitting of the simulation results and the analytical formulation is better for higher distances.

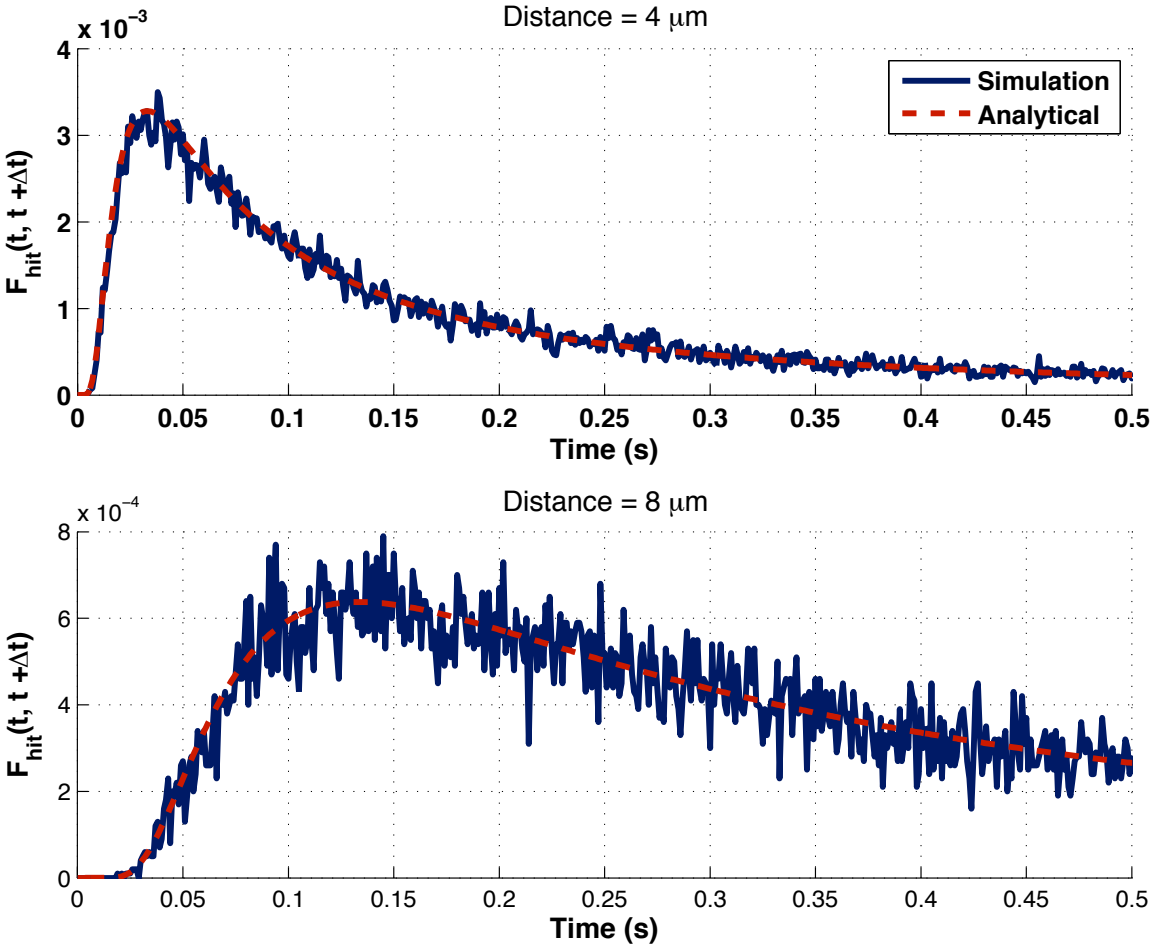


Figure 4.4. Analytical and simulation results for the fraction of molecules absorbed by the receiver versus time for $d = 4\mu m$ and $d = 8\mu m$.

4.3. The Number of Received Molecules

The expected number of molecules hitting the receiver in the interval $[t_1, t_2]$ for a given number of receptors can be evaluated by

$$\mathbb{E}[N^{\mathrm{Rx}}(t_1, t_2)] = \mathrm{N}^{\mathrm{Tx}}\ F_{\mathrm{hit}}(t_1, t_2), \tag{4.27}$$

where N^{Tx} denotes the number of emitted molecules at $t = 0$. The signal at a desired resolution, Δt, can be easily obtained by plotting the expected number of received molecules.

In Figure 4.5, the expected number of molecules hitting the receiver is depicted

for receiver radius values $5\mu m$, $10\mu m$, $20\mu m$, and $40\mu m$ and number of transmitted molecules $\mathrm{N}^{\mathrm{Tx}} = 100\,000$. For a given distance, as the radius increases, the expected number of received molecules increase as expected. The rate of increases is smaller on the other hand. This is because the hitting rate is proportional to r_r/r_0. When the receiver radius is high compared to the distance, the decrease on the expected number of received molecules as the distance increases becomes more linear. This is also expected since the effect of distance on r_r/r_0 will be smaller as r_r gets bigger.

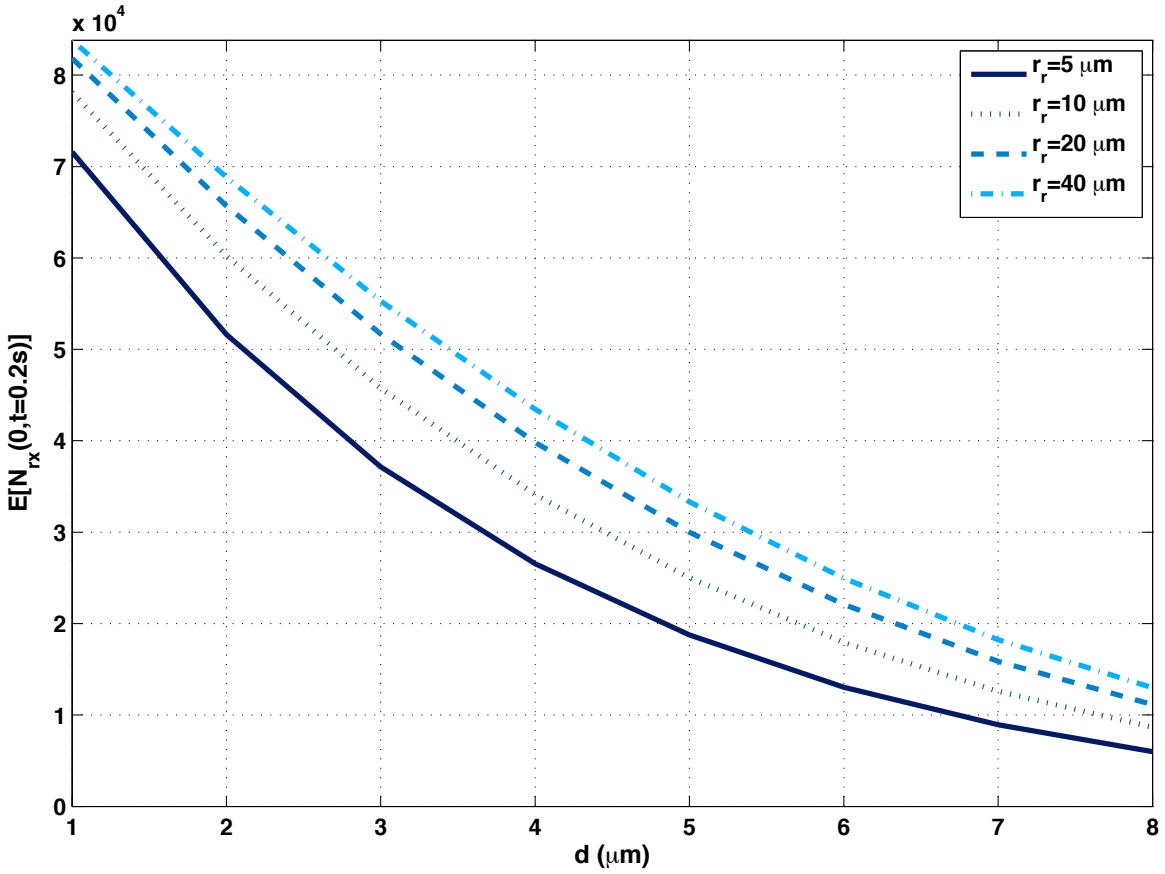


Figure 4.5. The number of received molecules till $t = 0.2s$ versus the distance for various receiver radius.

4.4. Chapter Summary

In this chapter, we consider the perfect reception process for an absorbing receiver in 3-D. We show the analytical formulation of the absorption rate and verify the result with simulations. We also investigate the effect of receiver size on absorption rate.

5. RECEPTION PROCESS FOR AN ABSORBING RECEIVER WITH RECEPTORS

The performance of molecular communication is significantly impacted by the reception process of the messenger molecules. The receptors' size and density, however, have yet to be investigated. In this chapter, we analyse the effect of receptor density and size on the signal reception of an absorbing receiver with receptors. We formulate the hitting rate of the molecules to the receptors of an absorbing receiver in a 3-D medium and verify our formulation via simulations. We also derive additional formulations to address receiver design issues. The results show that, when the total receptor area is the same, better hitting probability is achieved by using a higher number of relatively small receptors. In addition, deploying receptors, which cover a small percentage of the receiver surface, is able to create an effective communication channel that has a detectable signal level. Utilizing these results, it is possible to optimize the production costs of the receptors and the receivers.

5.1. Communication Model

The communication model used in this chapter is depicted in Figure 5.1. After being released from a transmitter, molecules propagate in their environment by following diffusion dynamics. While most scatter in the environment, some of these molecules, according to their type and the properties of the environment, reach the receiver. In the nature, a messenger molecule is received only when it binds to one of the receptors on the surface of the receiver. Then, for most receptor types, the messenger molecules are absorbed by the receiver. Therefore, each molecule contributes to the signal only once due to absorption or other mechanisms. For communication channel design, an important parameter is reception probability. The key factors affecting this parameter are the *size* and the *density* of the receptors. The number of receptors on the cell surface for signalling molecules can vary from 500 to (for a specific type of messenger molecule) more than 100 000 per cell [5]. If reception probability is too low,

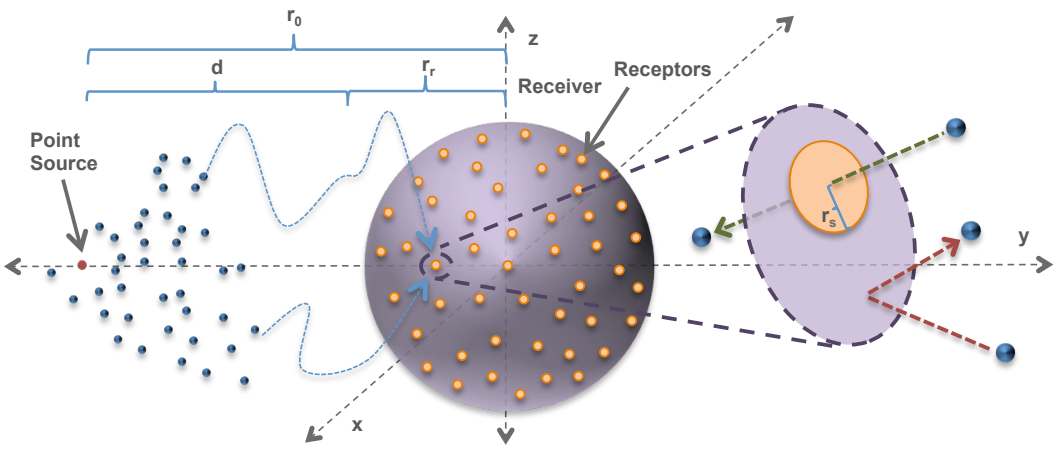
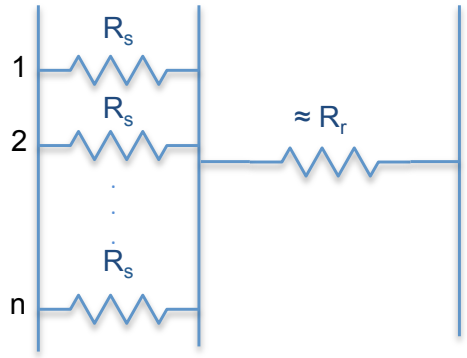


Figure 5.1. Communication model.

it may be impossible to establish an efficient communication channel; if it is unnecessarily high, it may denote inefficient use of the resources, specifically in terms of the energy and fragment molecules of the receptor. Reception probability should thus be well analyzed.

Messenger molecules are used as the information carriers between a point source and a spherical receiver with absorbing receptors. The point source is located at a distance r_0 from the center of the receiver. The point source and the spherical receiver both reside in a fluid propagation medium. It is assumed that the medium is unconfined, thus extending to infinity in all directions. After the information is modulated onto some physical property of the molecules, the molecules are released to the medium where they diffuse according to Brownian motion and arrive at the receiver. To absorb the molecules, the spherical receiver with radius r_r, uses its receptors with radius r_s. If a molecule collides with one of the n receptors deployed on the surface of the receiver, it is absorbed by the receiver. If it collides with the surface of the receiver without touching a receptor, it bounces back.



R_r: The diffusion resistance for a perfectly absorbing sphere
R_s: The diffusion resistance for a receptor

Figure 5.2. n-conductive patches model.

5.2. Absorption Rate of a Spherical Receiver with Absorbing Receptors

In the nature, a molecule is received by a receiver only when it binds to one of the receptors on the surface. To abstract this phenomenon, we model the receptors as circular areas over the receiver surface. A diffusing molecule is absorbed by the receiver only when it collides with a receptor. The other parts of the receiver surface are not capable of absorbing molecules. To model such a receiver, we need to derive the special case of absorption rate, where w depends on the number of receptors n, and the radius of receptors, r_s.

We start by solving (4.21) for arbitrary w and following the molecule distribution $p(r, t|r_0)$, hitting rate of the molecules to the receiver at time t can be obtained as [40],

$$f_{\text{hit}}(t) = \frac{r_r w}{r_0}\left(\frac{1}{\sqrt{\pi D t}}\exp\left[-\frac{(r_0-r_r)^2}{4Dt}\right] - \beta\exp\left[\beta(r_0-r_r)+\beta^2 Dt\right]\operatorname{erfc}\left[\frac{r_0-r_r}{\sqrt{4Dt}}+\beta\sqrt{Dt}\right]\right) \quad (5.1)$$

where $\beta = (wr_r + D)/(Dr_r)$. Furthermore, by integrating $f_{\text{hit}}(t)$ with respect to time,

we can obtain $F_{\text{hit}}(t)$ for arbitrary w, which is the fraction of molecules absorbed by the receiver until time t [40]:

$$F_{\text{hit}}(t) = \frac{r_r\beta - 1}{r_0\beta}\left(1 + \text{erf}\left[\frac{r_r - r_0}{\sqrt{4Dt}}\right] - \exp\left[(r_0 - r_r)\beta + Dt\beta^2\right]\text{erfc}\left[\frac{r_0 - r_r + 2Dt\beta}{\sqrt{4Dt}}\right]\right). \quad (5.2)$$

Using boundary condition for (5.2) as $t \to \infty$, we can calculate the fraction of received molecules for arbitrary w and steady state as

$$\lim_{t\to\infty} F_{\text{hit}}(t) = \frac{r_r\beta - 1}{r_0\beta}. \quad (5.3)$$

We can also formulate the fraction of molecules absorbed when $t \to \infty$ using an analogy with the electricity domain where n conductive patches are located on an insulating sphere, assuming $r_s \ll r_r$ [42]. The insulating sphere is analogous to the receiver and the receptors that bind with the molecules are analogous to the patches through which the current flows. For this scenario, the diffusion current I, which corresponds to the current in the electricity domain, is given by

$$I = C/R \quad (5.4)$$

where C is the concentration difference and R is the diffusion resistance. The diffusion resistance for a sphere with absorbing receptors R can be written as

$$R = R_r\left(1 + \frac{\pi r_r}{n r_s}\right) \quad (5.5)$$

where R_r is the diffusion resistance of a perfectly absorbing sphere [42]. This equation shows that the diffusion resistance of a receiver with receptors is larger than that of a perfectly absorbing sphere by a factor of $1 + (\pi r_r)/(n r_s)$. Using (5.4) and (5.5), for the

steady state, we can write

$$\frac{I}{I_r} = \frac{1}{1 + \frac{\pi r_r}{n r_s}} \tag{5.6}$$

where I and I_r are the diffusion current for a sphere with absorbing receptors and a perfectly absorbing sphere. The fraction of molecules absorbed for an absorbing sphere at $t \to \infty$ is r_r/r_0, hence we can write the fraction of molecules for a sphere with absorbing receptors as

$$\lim_{t\to\infty} F_{\text{hit}}^{r_s,n}(t) = \frac{r_r}{r_0}\frac{1}{1 + \frac{\pi r_r}{n r_s}} = \frac{r_r}{r_0}\frac{r_s n}{r_s n + \pi r_r} \tag{5.7}$$

At the boundary condition where $t \to \infty$, (5.3) and (5.7) will be equal. Using this equality, we can write w and β as,

$$w = \frac{n r_s D}{\pi r_r^2}, \quad \beta = \frac{n r_s + \pi r_r}{\pi r_r^2} \tag{5.8}$$

Using (5.8) and (5.2), we can derive the formula for a receiver with absorbing receptors as [58],

$$\begin{aligned} F_{\text{hit}}^{r_s,n}(t) = \frac{r_r}{r_0}\frac{r_s n}{r_s n + \pi r_r}\Bigg(1 + \text{erf}\left[\frac{r_r - r_0}{\sqrt{4Dt}}\right] \\ - \exp\left[(r_0 - r_r)\left(\frac{n r_s + \pi r_r}{\pi r_r^2}\right) + Dt\left(\frac{n r_s + \pi r_r}{\pi r_r^2}\right)^2\right] \\ \times \text{erfc}\left[\frac{r_0 - r_r + 2Dt(\frac{n r_s + \pi r_r}{\pi r_r^2})}{\sqrt{4Dt}}\right]\Bigg). \end{aligned} \tag{5.9}$$

Using (5.9), we define the fraction of molecules received between t_1 and t_2 as

$$F_{\text{hit}}^{r_s,n}(t_1, t_2) = F_{\text{hit}}^{r_s,n}(t_2) - F_{\text{hit}}^{r_s,n}(t_1). \tag{5.10}$$

To analyse the formulations, we use a simulation scenario where the radius of the

receiver is $10\mu m$, radius of the receptors are $0.04\mu m$, number of receptors is 1250, the diffusion coefficient is $79.4\mu m^2/s$, and time step is $10^{-6}s$. The simulation is repeated for 100 000 molecules to calculate the hitting rate via simulation.

Figure 5.3 shows the simulation results and analytical formulation for $d = 1\mu m$, which means the point source is located at a distance of $r_0 = 11\mu m$ from the centre of the receiver. The results are depicted as a histogram with granularity of 10^{-3}. The simulation results and the analytical results overlap with a deviation where the peak is observed. The reason of this is due to the relatively large time step of $10^{-6}s$. Some of the molecules are not captured due to large time step as described in Section 3.3 when the receiver has small receptors deployed over it surface.

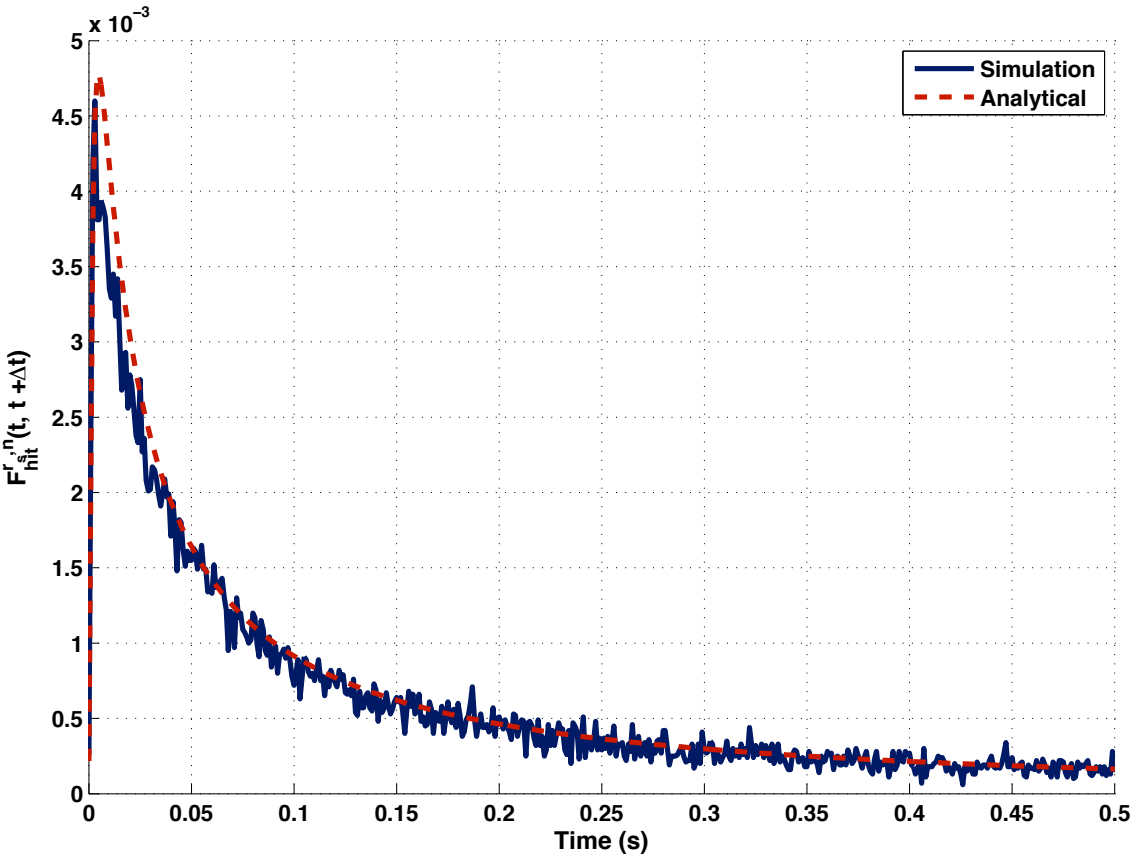


Figure 5.3. Analytical and simulation results for the fraction of molecules absorbed by the receiver versus time for $d = 1$.

In Figure 5.4 the simulation results and analytical formulation are shown for $d = 2\mu m$, which means the point source is located at a distance of $r_0 = 12\mu m$ from the centre of the receiver. Since the distance is higher, the hitting rate is smaller, and

molecules scatter around more. Hence, the variation of the simulation results becomes a bit higher. Again a deviation at peak value is observed due to the relatively large time step for an implementation with receptors. Some of the molecules are not captured due to large time step.

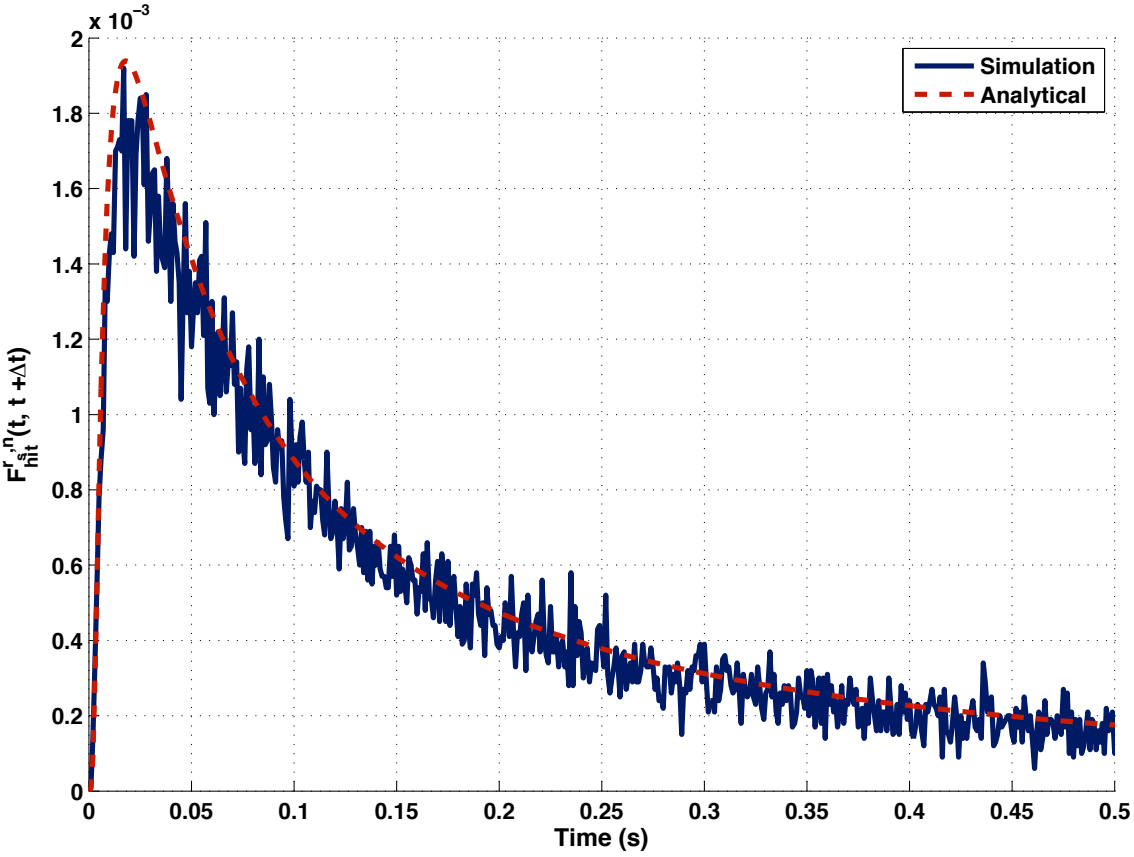


Figure 5.4. Analytical and simulation results for the fraction of molecules absorbed by the receiver versus time for $d = 2$.

Figure 5.5 shows the simulation results and analytical formulation for $d = 4\mu m$ and $d = 8\mu m$, which means the point source is located at a distance of $r_0 = 14\mu m$ and $r_0 = 18\mu m$ from the centre of the receiver respectively. As the distance increases, the time to peak value increases similar to the receiver without receptors. The molecules also scatter to the environment more since the hitting rate also decreases. This is why the variation of the simulation results increases as well. For higher distances, the effect of time step becomes smaller since less molecules are able to reach to the proximity of the receiver. This is why the fitting of the simulation results and the analytical formulation is better for higher distances.

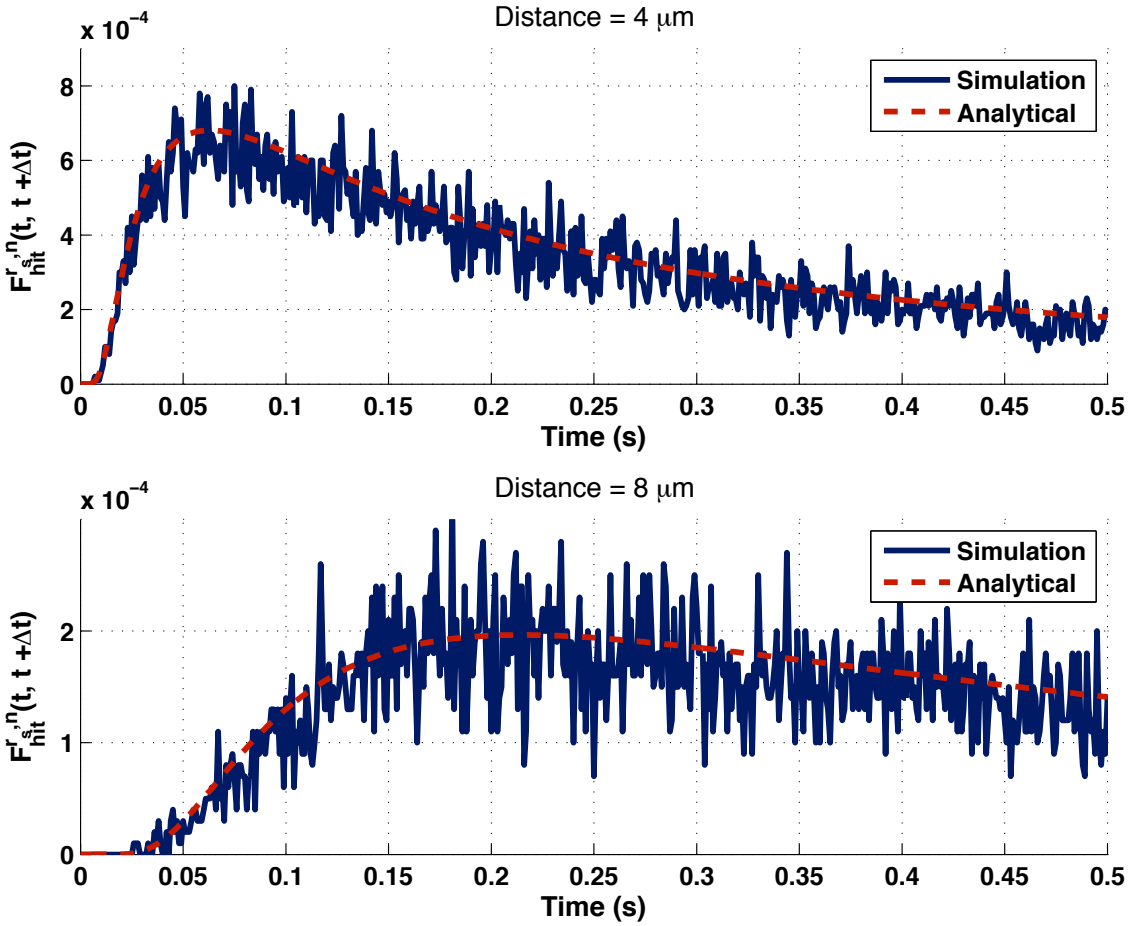


Figure 5.5. Analytical and simulation results for the fraction of molecules absorbed by the receiver versus time for $d = 4$ and $d = 8$.

5.3. The Number of Received Molecules

The expected number of molecules hitting the receiver in the interval $[t_1, t_2]$ for a given number of receptors can be evaluated by

$$\mathbb{E}[N^{\mathrm{Rx}}(t_1, t_2)] = \mathrm{N}^{\mathrm{Tx}}\ F_{\mathrm{hit}}^{r_s,n}(t_1, t_2), \tag{5.11}$$

where N^{Tx} denotes the number of emitted molecules at $t = 0$. The signal at a desired resolution, Δt, can be easily obtained by plotting the expected number of received molecules.

Figure 5.6 shows the number of received molecules until $t = 0.2s$ versus distance

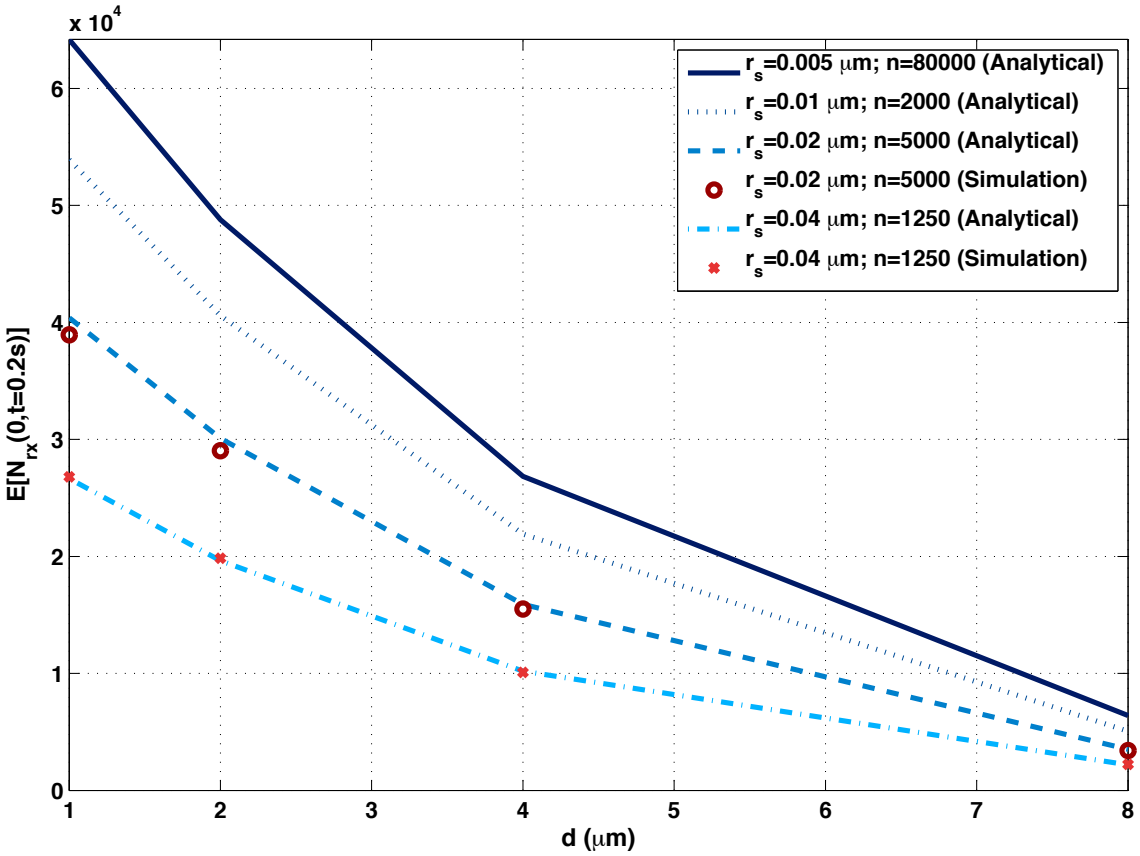


Figure 5.6. The number of received molecules till $t = 0.2s$ versus the distance for the same total receptor area of 2π ($N^{Tx} = 100\,000$).

d where the total area that is covered by the receptors is kept constant for different r_s and n values. $r_s = \{0.04,\, 0.02\}\,\mu m$ cases are simulated and the simulation results are coherent with the analytical results. The results indicate guidance for an important receptor design criteria. For any fixed distance, given the total area that will be covered by the receptors, to achieve a better hitting probability, one should use a higher number of relatively small receptors. The achieved nominal gain gets smaller as the distance increases.

5.4. Receptor Area Analysis

To create an efficient communication channel, once the appropriate receptor type has been selected, it is important to find the minimum sufficient ratio of the total surface area that should be covered with receptors. This decision has a direct effect on receiver production costs.

To analyse this, we should have a formula or a method to find the minimum number of receptors needed to achieve a specific $F_{\text{hit}}^{r_s,n}(t)$ value, α, for the given parameters, which is denoted as n_α. Note that, when r_0, r_r, r_s, D, t are fixed, $F_{\text{hit}}^{r_s,n}$ strictly increases as the number of receptors increase as shown in Figure 5.7 for $r_r = 10\mu m$, $r_0 = 11\mu m$, $t \to \infty$, and $D = 79.4\mu m^2/s$. Therefore, for a given α, the set $\{n|F_{\text{hit}}^{r_s,n} = \alpha\}$ has a single element, which is n_α. Since $F_{\text{hit}}^{r_s,n}$ strictly increases with respect to n, it enables us to perform a numerical search using (5.9) to find n_α.

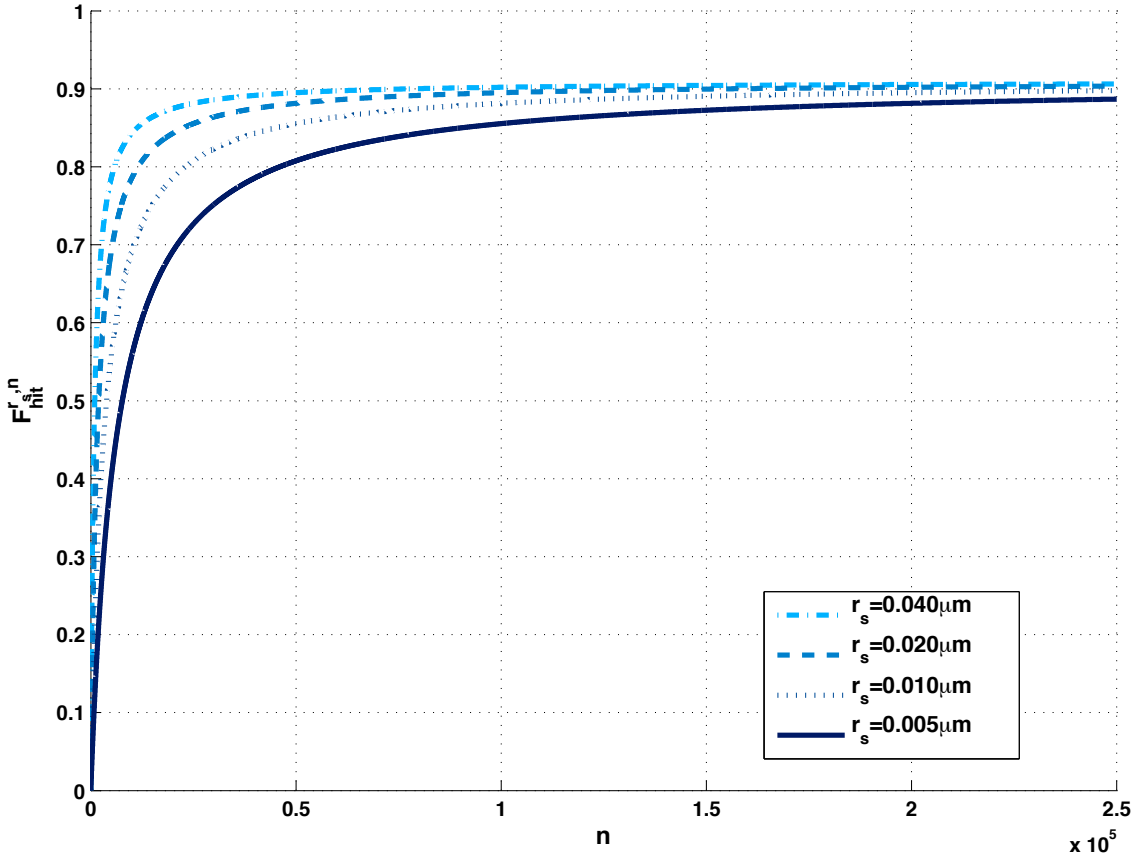


Figure 5.7. Fraction of molecules absorbed by the receiver versus the number of receptors for different r_s values.

The ratio of the total area of the receptors to the total area of the perfectly absorbing receiver can be written as

$$\frac{A^{r_s,n}}{A} = n\left(\frac{r_s}{2r_r}\right)^2 \tag{5.12}$$

where $A^{r_s,n}$ is the surface area covered by the receptors and A is the total surface area of the receiver. Finally, using n_α and (5.12), we can calculate the ratio of the surface

area to be covered by receptors to achieve $F_{\text{hit}}^{r_s,n} = \alpha$ using

$$A_\alpha = n_\alpha \left(\frac{r_s}{2r_r} \right)^2 . \tag{5.13}$$

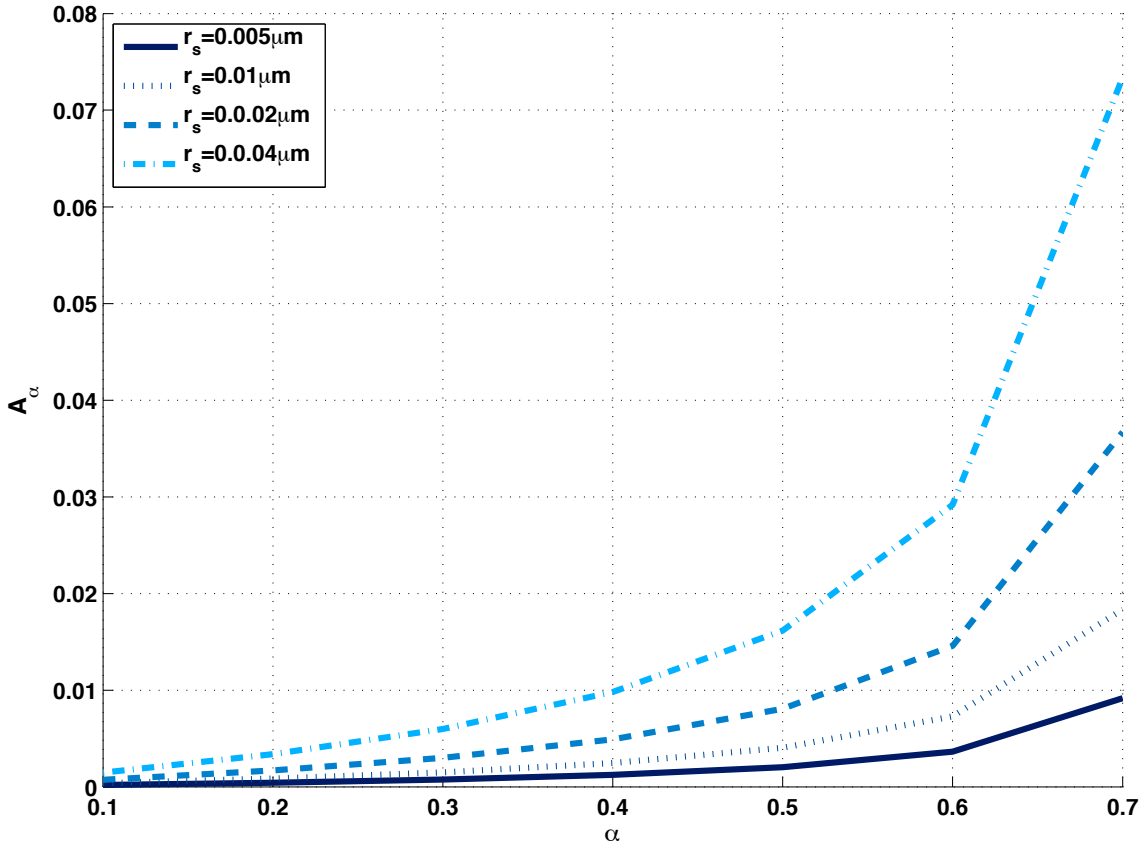


Figure 5.8. Ratio of the surface area to be covered by receptors to achieve $F_{\text{hit}}^{r_s,n}(t) = \alpha$.

Figure 5.8 illustrates A_α versus α for $t = 0.2s$, $r_r = 10\mu m$, $r_0 = 11\mu m$, and $D = 79.4\mu m^2/s$. The results show that, to achieve significant $F_{\text{hit}}^{r_s,n}(t)$ values, it is sufficient to deploy receptors to cover only a miniscule ratio of the total surface area. For instance, for $r_s = 0.005\,\mu m$, the ratio of the receptor area to the full surface area so as to achieve $\alpha = 0.7$ is 0.0092. Hence, it is possible to achieve $F_{\text{hit}}^{r_s,n}(t) = 0.7$ by covering less than 1% of the total surface area of the receiver, where $F_{\text{hit}}(t) = 0.7811$ for perfectly absorbing sphere. This shows that it is possible, on a practical level, to deploy receptors for several different molecule types and achieve considerable signal energy for each communication channel that uses different molecule types.

5.5. Receptor Deployment

The deployment of the receptors over the receiver surface is another parameter that affects the received signal. In this section, we discuss the heterogeneity of the receptors in the nature and the effect of such deployment on the received signal.

5.5.1. Receptor Heterogeneity in Nature

Epithelial cells, neurons, and migrating cells are examples of polarized cells that have heterogeneous receptor deployments that exist in the nature. For receiving different types of signals, a cell has different types of receptors over the different parts of the cell membrane. Heterogeneous receptor deployment is an adaptation to the environment and the signalling mechanism. Also, the density of receptors varies according to the environment and the adaptation mechanisms.

Neuron is one of the extensively studied polarized cells. The γ-Aminobutyric Acid (GABA), inhibitory neurotransmitter and glutamate as a major excitatory neurotransmitter, regulates the excitability of the neurons. Ionotropic receptors for both neurotransmitters are heterogeneous on the cell membrane.

The γ-Aminobutyric Acid type A ($GABA_A$) receptor is an integral membrane protein complex that is widely distributed in the brain, mediating the main synaptic actions of GABA [59]. $GABA_A$ receptors belong to the molecular superfamily of ligand-gated ion channels. The receptors respond to the presence of GABA by the opening of an intrinsic anion channel [60]. Due to the existence of $GABA_A$ receptor subunits and their combinations, there exist multiple $GABA_A$ receptors in the brain that show differential distribution, differences in affinity, and distribution of binding sites for benzodiazepine (BZD) receptor ligands and in their allosteric modulation properties [61, 62]. Subunit repertoire consists of 19 different types; $\alpha 1-6$, $\beta 1-3$, $\gamma 1-3$, δ, ε, θ, π, and $\rho 1-3$ [63]. Depending on their subunit composition, $GABA_A$ receptors have specific anatomical distributions including subcellular localization [64].

Glutamate receptors are counteracted mainly by GABA activated anion channels. These receptors are found in different locations of the cell membrane and validate the receptor heterogeneity. Glutamate induces a depolarization of the cell membrane by activating related receptors. When depolarization reaches a certain level, glutamate stimulus induces an action potential [60]. A concurrent inhibitory signal mediated by GABA hinders the development of a threshold depolarization and the action potential. One side of the cell contains $GABA_A$ receptors and the other side contains glutamate receptors. Therefore, different signals are received via different regions of the neuron cell.

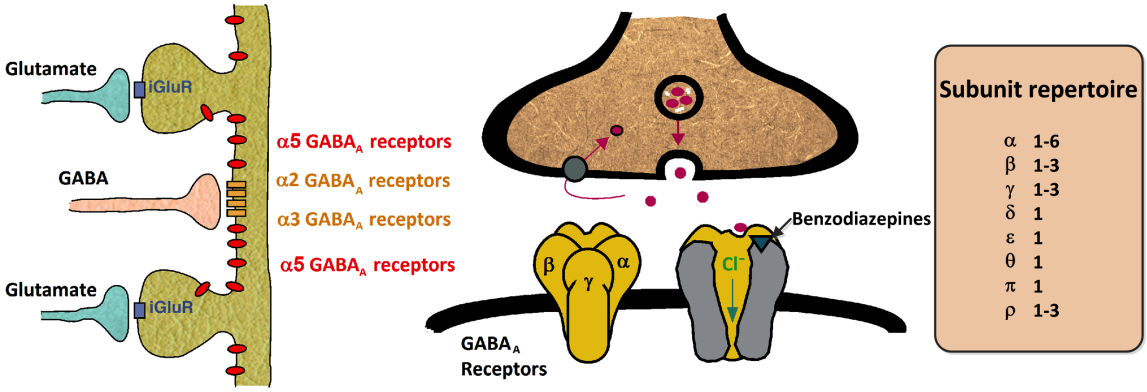


Figure 5.9. Distribution of $GABA_A$ receptor subtypes on a hippocampal pyramidal cell dendrite and the subunit repertoire of $GABA_A$ receptors (adapted from [1]).

In Figure 5.9, distribution of $GABA_A$ receptor subtypes and ionotropic glutamate receptors (iGluR) is depicted. Extraordinary heterogeneity in the distribution of $GABA_A$ receptor subunits is observed. The $GABA_A$ $\alpha 2$, $\alpha 3$, and $\alpha 5$ subunits show a differential subcellular distribution. The $\alpha 2$ and $\alpha 3$ subunits appear at GABAergic synaptic regions whereas the $\alpha 5$ subunit is confined to extrasynaptic regions.

5.5.2. Effect of Receptor Heterogeneity

In all previous scenarios, the receptors are deployed all over the receiver nanomachine surface. To analyse the effect of different deployment scenarios that are also observed in the nature, as described in Section 5.5.1, we simulate five different cases for different transmission distances to analyse $F_{\text{hit}}^{r_s,n}(t)$. For all scenarios, the number of receptors is 80 000, and the receptor radius is 0.005 μm and simulated time is 100

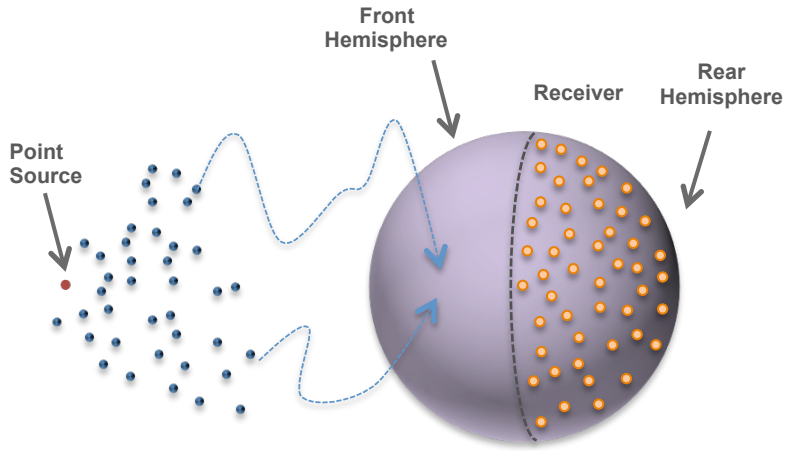


Figure 5.10. Deployment of receptors when the percentage of the receptors at the front hemisphere is zero.

seconds. The details of the scenario is shown in Figure 5.10. In the first case, all of the receptors are deployed on the front hemisphere of receiving nano-machine that is towards the transmission point. In the second case, two thirds of the receptors are deployed towards transmission point, and one third is deployed on the rear hemisphere of the receiver nano-machine. In third scenario, receptors are uniformly distributed over both hemispheres. In the fourth scenario, one third of the receptors is deployed on the front hemisphere towards the transmission point and two thirds are on the rear hemisphere. Lastly, in the fifth scenario, all receptors are deployed on the rear hemisphere that is not facing the transmission point. The simulation is repeated for 10 000 molecules to calculate the probability of hit.

The results are shown in Figure 5.11. A substantial increase in the probability of hit is observed when the receptors are moved from rear hemisphere to front hemisphere, specially until the receptors are homogeneously distributed. After that point, some increase in the probability of hit is observed for small distances, while the effect gets smaller as the distance increases. This can be explained by the nature of random walk. For smaller distances, it is not likely that the messenger molecules go around the receiving nano-machine and hit a receptor at the rear hemisphere. As the distance increases, this probability also increases. We can conclude that the transmission distance should also be considered for the optimal receptor deployment model.

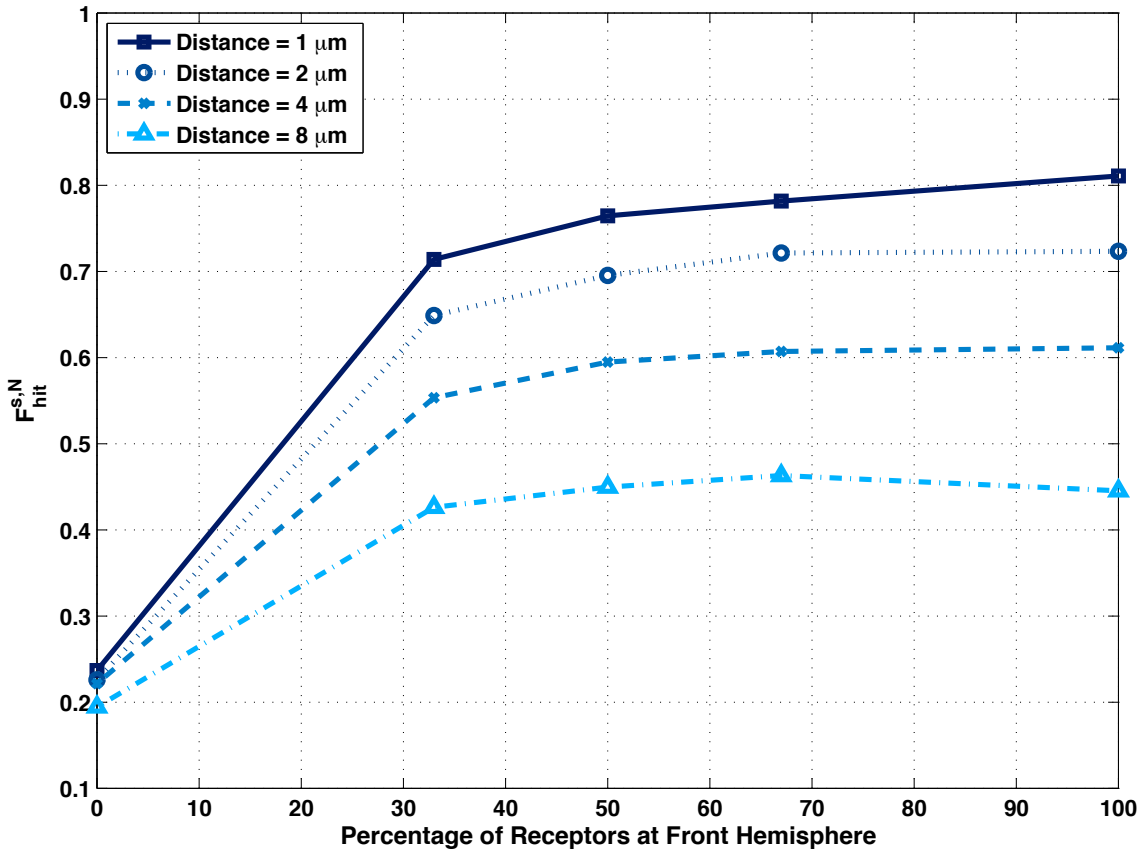


Figure 5.11. Hitting rate versus receptor heterogeneity for different distances.

5.6. Chapter Summary

In this chapter, we consider the imperfect reception process in nature to build a more realistic model and derive an analytical formulation of the absorption rate of a spherical receiver with absorbing receptors in a 3-D medium. We also addressed receiver design issues, specifically, the optimization of the size and the total area of the receptors. This has a direct effect on the production costs of the receptors and receivers. We used the formulations to conclude that it is possible, at a practical level, to have a comparable signal energy to a perfectly absorbing sphere while covering as little as 1% of the surface area of the receiver. We also show the effect of receptor heterogeneity on the hitting probability.

6. DEMODULATION OPTIONS FOR AN ABSORBING RECEIVER

In a generic molecular communication model, the aim is to modulate information, and transmit it to a propagation medium by a transmitter nanomachine, which is then received by a receiving nanomachine and demodulate to access the information. All current research effort is concentrating on digital modulation options in their models. In this chapter, we introduce another approach in which data to be transmitted is not quantified, but instead directly sent via molecular diffusion [65]. This approach will enable higher data rates for specific set of applications which can tolerate bounded error and is limited to specific set of information flow.

It should be noted that in this chapter, we use the term "analog" to describe the fact that, similar to analog communication, the value to be transmitted is not digitized, but the actual number of molecules transmitted is not continuous but discrete.

6.1. Digital Molecular Communication Model

In a digital molecular communication model, digital modulation is used. This model can be seen as a direct implementation of the current digital communication model to nano-scale. With this approach, a binary representation of the information is delivered as ones and zeros to the receiving party. So during each symbol duration, binary information is sent. Another important point to note is the effect of channel error. In case of an error, any of the transmitted bits can be corrupted with equal probability, but based on the position of the error, the actual data can be demodulated to a value with unbounded error. This approach requires additional error checking and correction, or retransmission to make sure that correct value is transmitted, which also decreases overall channel capacity. The advantage of this approach is that, by utilizing error correction mechanisms, receiver can be sure that it gets exactly the same data sent by the transmitter. This requirement may be crucial for some applications, such as transmission of a command on a DNA code to be executed. However, there may be

cases where the exact transmission of the data may not be as important as the data rate achieved.

6.2. Analog Molecular Communication Model

In the analog communication model, each nano-machine sends information not via digital modulation. Instead, during each symbol duration the nano-machine sends specific number of molecules that correspond to the information to be sent. The same structure and mechanisms are used as in the digital model, but the value is directly sent with corresponding concentration level instead of binary encoding. Following steps are repeated at each symbol duration.

- **Encoding**: Before each symbol duration a vesicle that contains same number of molecules, scaled according to the value of the data, is prepared.
- **Transmission**: These molecules are pumped to the propagation medium at start of the symbol duration.
- **Propagation**: The molecules diffuse through the propagation medium. Since it is not a guided transmission medium, the molecules follow the physical characteristics of the medium for propagation.
- **Reception**: Throughout the symbol duration, the molecules that reach the receptors of the receiving nano-machine are successfully received. Each nano-machine has specific receptors, so that only certain molecules are accepted into the nano-machine.
- **Decoding**: Based on the number of molecules received during the symbol duration, and previous symbols, the receiver estimates the number of molecules sent by the receiver.

The advantage of the analog approach over digital transmission is shorter symbol durations, lower energy consumption, and better noise resilience.

A data item that is composed of X bits when digitized consumes X symbol durations while the same content can be transmitted in one symbol duration and

consume less energy, at the expense of some bounded error in transmission. It should also be noted that the analog communication is limited to a subset of possible input types. Given these characteristics, analog transmission is suitable for some set of communication needs, while digital transmission is suitable for others.

6.3. Energy Consumption and Data Rate

A mathematical model can be used to analyze energy requirements and data rate for the analog and the digital communication. In both cases, we are interested in the energy and time required to transmit n molecules in a time slot. For the digital method, this corresponds to the transmission of one or few bits (depending on modulation), while in the analog model, the whole value is transmitted. We follow the generic energy model described in [20]. We use the same definitions to model the power consumption in each step in the release of a vesicle;

- E_S: Synthesis of the messenger molecules from their building blocks
- E_V: Production of a secretory vesicle
- E_C: Carrying the secretory vesicles to the cell membrane
- E_E: Releasing the molecules via the fusion of the vesicle and the cell membrane

The total energy consumption E_T to release m molecules is given, accordingly, by

$$E_T = mE_S + \lceil \frac{m}{c_v} \rceil (E_V + E_C + E_E) \tag{6.1}$$

where c_v is the capacity of a vesicle in terms of the messenger molecules.

6.3.1. Transmission of an integer value

The model defined above can be used to investigate the energy consumption and time required to transmit an integer value X from transmitter to receiver using the digital and the analog models. It is assumed that the integer value is encoded as n

bits.

6.3.1.1. Digital Transmission. In digital transmission, X is transmitted in $n{\cdot}t_s$. During each t_s, either a vesicle with m_{max} molecules is transmitted or no transmission is done, based on the value of the bit for corresponding t_s, as shown in Figure 6.1.

To find the expected value of energy consumed to send an arbitrary integer value, let us define P_i as the probability of sending a vesicle with m_{max} molecules at t_{s1}. Then, the expected value of the consumed energy, E_{TOTAL}, is

$$E[E_{TOTAL}] = \sum_{i=0}^{n} P_i \cdot E_T \tag{6.2}$$

For a random input, P_i will be $\frac{1}{2}$ for $i = 1..n$, and substituting E_T, we have

$$E[E_{TOTAL}] = \frac{n}{2}(m_{max}E_S + \lceil\frac{m_{max}}{c_v}\rceil(E_V + E_C + E_E)) \tag{6.3}$$

One should note that the energy consumption is proportional to m_{max}. In [20], it is shown that if the distance between the transmitter and the receiver increases, then t_s and m_{max} also need to be increased to achieve high capacity, therefore, to maintain the same capacity level, energy consumption also increases as the distance between the nano-machines increases.

6.3.1.2. Analog Transmission. In the analog transmission, X is transmitted during a single t_s, in which a vesicle with X molecules will be transmitted as shown at Figure 6.1.

To find the expected value of energy consumed to send an arbitrary integer value X, where X is uniformly distributed between 0 and m_{max}, let's define Px_i as the

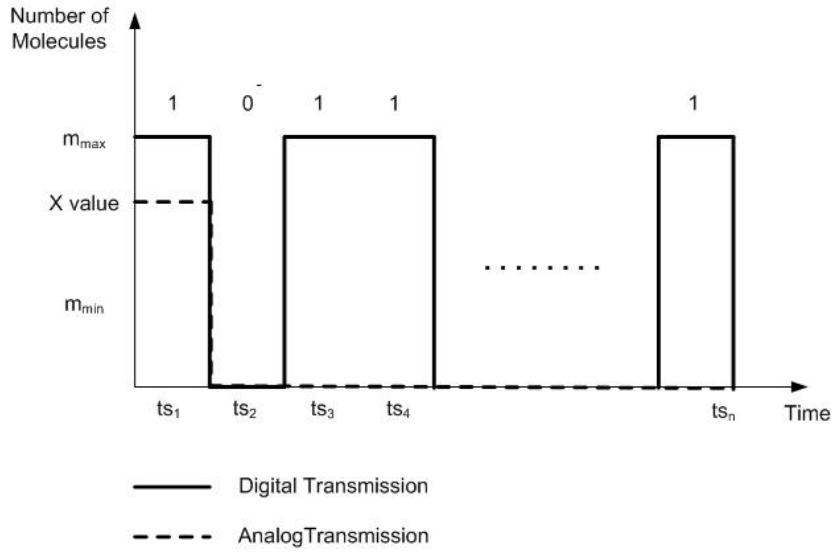


Figure 6.1. Molecules sent in digital and analog transmission.

probability of sending value i. Then, $E[X]$ is

$$E[X] = \sum_{i=0}^{m_{max}} Px_i \cdot i \tag{6.4}$$

Since X is uniformly distributed, we can write

$$E[X] = \frac{1}{m_{max}+1}.\frac{m_{max}(m_{max}+1)}{2} = \frac{m_{max}}{2} \tag{6.5}$$

We can calculate expected value of energy consumption as

$$E[E_{TOTAL}] = \frac{m_{max}}{2}E_S + \lceil\frac{m_{max}}{2c_v}\rceil(E_V + E_C + E_E) \tag{6.6}$$

The time required to transmit same data in analog transmission is $\frac{1}{n}$ of the digital method.

6.4. Results

The performance of the proposed method is analysed by means of the simulators described in Section 3. A transmitter and a receiver nano-machines in a liquid envi-

ronment are simulated where the molecules propagate according to Brownian motion. In the simulations, the radius of the transmitter and receiver is $10\mu m$ and diffusion coefficient is 79.4 $(\mu m)^2/s$. Simulations are repeated ten times to calculate the average results.

6.4.1. Communication Channel without Noise

We simulate the transmission of a chunk of information from one nano-machine to another using both the digital and the analog communication methods subject to the distance and symbol durations. Appropriate symbol duration values for communication over different distances are taken from [20], which are calculated using average hitting times of the molecules in simulation.

For the sake of simplicity and generality, we assume that this information chunk is one byte. In the digital communication method, the value of the byte value is sent as a sequence of eight bits during eight symbol durations, where binary "one" is represented with 128 molecules. On the contrary, in the analog communication method the number of molecules is equal to the byte value sent, which can be at most 255. With these parameters, the analog communication method can send the same information eight times faster, using four times less energy. For better comparison, we analyze the results where the symbol durations and the energy levels are taken to be equal for both methods, and symbol durations increases in parallel to distance both in digital and analog version. The results are depicted in Figure 6.2. For small symbol durations, the digital communication method is prone to errors, whereas the analog communication method operates with much smaller error rates. As the distance increases, since symbol duration increases dramatically, the performance of the digital communication also increases. Note that after 16 micrometers, it is not possible to receive adequate number of molecules at the receiver; hence communication is not feasible.

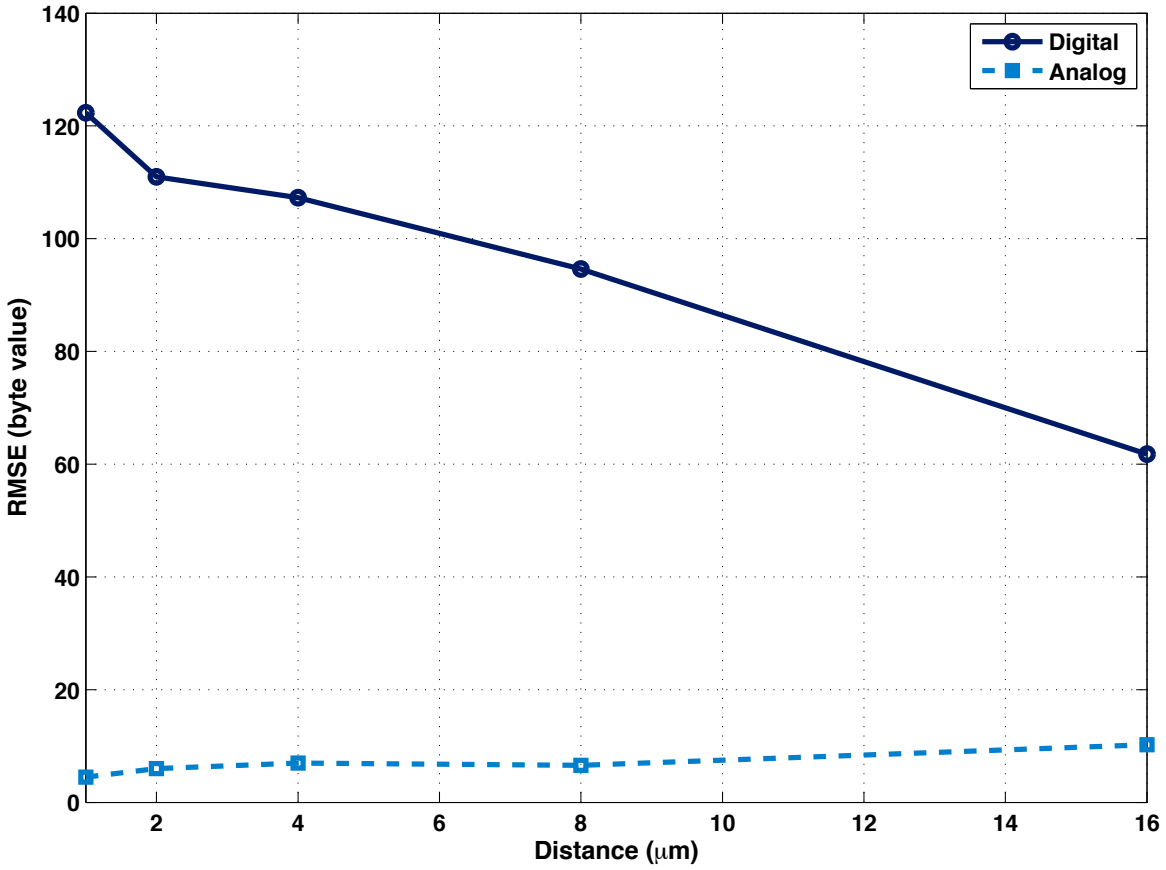


Figure 6.2. Analog vs Digital.

6.4.2. Communication Channel with Noise

To show the behavior of the proposed method in a communication channel with noise, we consider White Gaussian Noise where molecules are inserted to or removed from the medium. For this purpose, at each time interval a random variable with zero mean and corresponding variance for different noise levels is generated. This value can be a negative or positive. Based on the value of the random variable, molecules are removed from or inserted to the simulation environment. The results for the digital and the analog communication are shown in Figure 6.3. Both methods experience similar effects under noise, and the error rate increases at a comparable rate.

It is apparent from Figures 6.2 and 6.3 that the digital communication does not emerge as a feasible alternative at such noise levels and symbol durations. Therefore, we delve the operational range of the digital communication. Figure 6.4 shows the

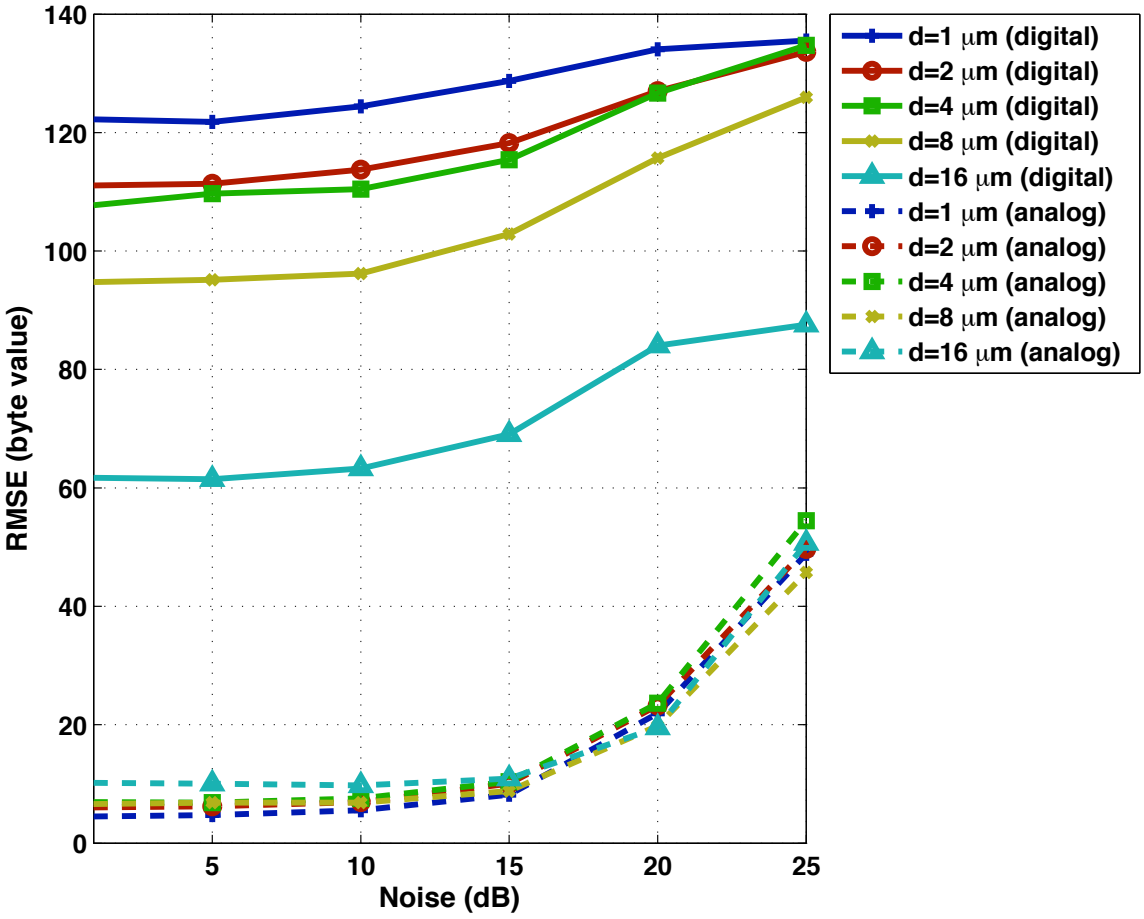


Figure 6.3. Noise Effect with Same Durations.

results where the analog and the digital communication use the same energy level, i.e., the same number of bits are transmitted in both cases. However, compared to the analog communication, it takes eight times longer for the digital communications to send one byte of information. When there is no noise, the digital communication can send the data without any error, and the analog communication experiences errors due to its communication mechanism. As the noise level increases, the digital method starts to experience high error rates, whereas the analog method is more resilient to noise. Thus, the analog method outperforms the digital method if the data needs to be sent in a shorter time or if the noise in the channel is expected to be high.

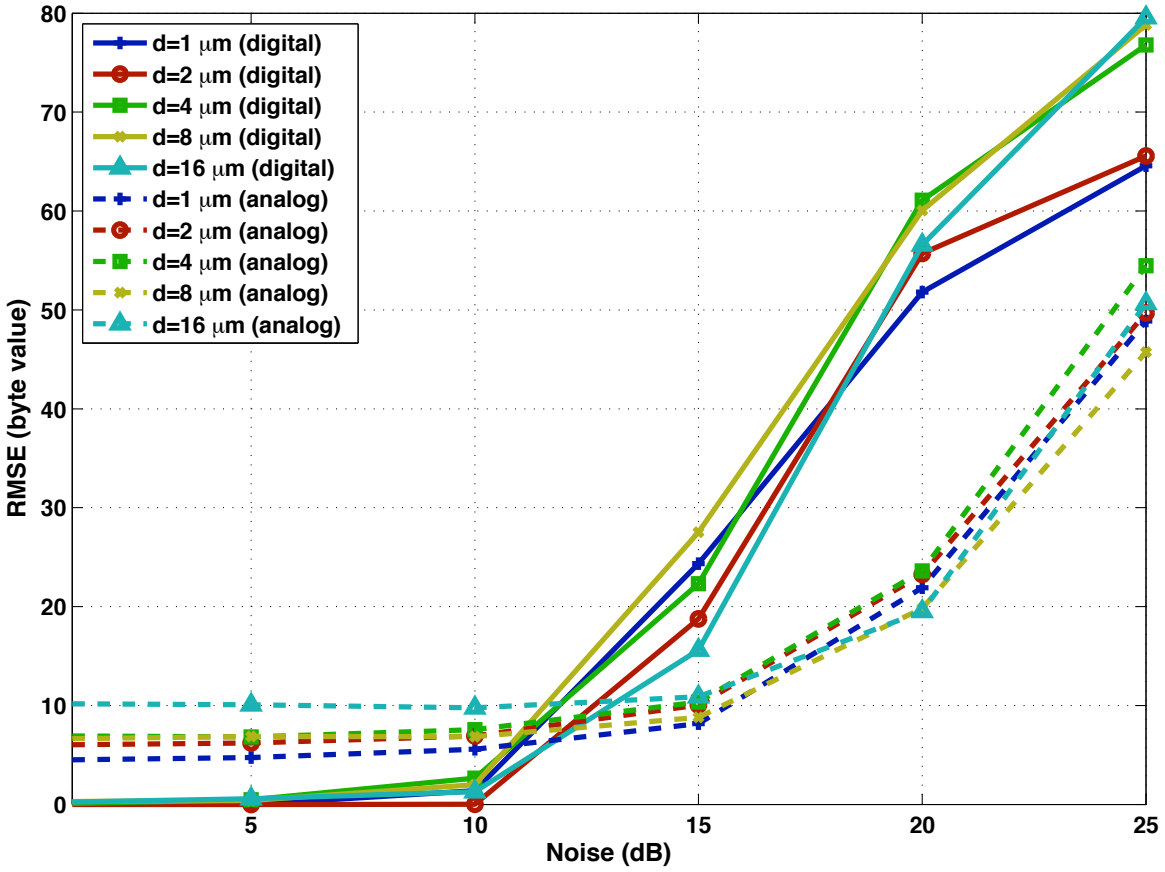


Figure 6.4. Noise Effect with Different Durations.

6.5. Chapter Summary

In this chapter, we consider the analog communication as an alternative to the digital communication for specific applications where higher data rate and lower energy consumption can be traded for limited errors in transmission. We show that, for the scenarios that use specific range of data, the analog communication emerges as a viable alternative to its digital counterpart by providing faster and more energy efficient communications with bounded errors.

7. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In this thesis, firstly, we introduce two simulation models for effective simulation of molecular communication. The first simulation model uses dual-zone model to decrease simulation execution time while preserving accuracy. We show that as the number of molecules in the environment increases, the gain obtained thanks to the proposed model also increases. A possible future research direction is to extend the dual-zone approach to multi-zone and analyse the effect on speed up and accuracy. Second simulation model we propose is an HLA based simulation architecture for molecular communication. The architecture enables defining inter-operable and reusable simulation components, which can be executed in a distributed environment with the possibility of interacting with live systems. This makes the execution of large scale hybrid scenarios possible. In these scenarios, simulation and live system components distributed to different geographical locations can interact with each other, which will be very beneficial for future research activities. The architecture components are described and the performance of the simulation tool is demonstrated using a simulation scenario. It is shown that if computationally intensive collision algorithms, transmission or reception algorithms, or molecule movement algorithms need to be analysed, the benefits of distributed architecture is fully utilized.

Utilizing the proposed simulation infrastructure, we consider the imperfect reception process in nature to build a more realistic model and drive the analytical formulation of the hitting rate to the receptors of a spherical absorbing receiver in 3-D medium. We also provide additional formulations to address receiver design issues, specifically, optimization of the size and the total area of the receptors, which will have a direct effect on the production costs of the receptors, and the receivers. As an extension of the current work, analytical models for heterogeneity and different receptor shapes can be considered.

Finally, we propose an alternative demodulation method for absorbing receivers. We consider the analog communication as an alternative to the digital communication. The model provides higher data rate and lower energy consumption for communication of specific type of input in exchange for bounded errors. We believe such applications will exist in environments where many transmitter and receiver nano-machines try to communicate using molecules in a noisy environment. Although the analog method does not deliver the data as accurate as the digital method in a perfect channel without any noise, it is shown that same data can be sent with much lower energy and in much shorter time frame as soon as reasonable amount of noise appears. It is also shown that the analog method is more resilient to noise. The decision on which method to use highly depends on the requirements. For applications that can afford bounded error but require high data rates and energy efficiency, the analog method seems to be a promising alternative.

APPENDIX A: SOURCE CODE for 1-D DIFFUSION

This chapter includes a sample implementation of Brownian motion for the simulation of particles in one dimension using Java programming language. A graphical user interface shows how the distribution of the molecules changes at specific locations.

A.1. Diffusion1D Class

```
package tr.edu.boun.cmpe.diffusion1D;

import javax.swing.JFrame;
import javax.swing.WindowConstants;

public class Diffusion1D {

        public static void main(String[] args) throws Exception {

                double currentTime = 0;
                PropagationEnvironment propagationEnvironment = new
                   PropagationEnvironment();

                JFrame frame = new JFrame("Diffusion 1D");
                frame.setSize(600, 600);
                frame.setDefaultCloseOperation(WindowConstants.
                   EXIT_ON_CLOSE);
                Diffusion1DGUI gui = new Diffusion1DGUI(
                   propagationEnvironment);
                frame.add(gui);

                frame.setVisible(true);

                while (currentTime < SimulationParameters.SIMULATION_TIME
                   ) {
                        propagationEnvironment.advanceTime();
                        currentTime += SimulationParameters.TIME_STEP;
```

```
                        frame.repaint();
                        Thread.currentThread().sleep(SimulationParameters
                            .SLEEP_TIME_FOR_GUI);

                }
                propagationEnvironment.print();
        }
}
```

A.2. Diffusion1DGUI Class

```
package tr.edu.boun.cmpe.diffusion1D;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Ellipse2D;

public class Diffusion1DGUI extends javax.swing.JPanel {

        private PropagationEnvironment propagationEnvironment;

        public Diffusion1DGUI(PropagationEnvironment pe) {
                propagationEnvironment = pe;
                initComponents();
        }

        public void paint(Graphics g) {
                Graphics2D g2 = (Graphics2D)g;
                g2.setColor(Color.darkGray);
                g2.drawLine(0, 550, this.getWidth(), 550);
                Molecule[] molecules = propagationEnvironment.
                    getMolecules();
                g2.setColor(Color.red);
                for (int i = 0; i < molecules.length; i++) {
                        g2.fillOval(this.getWidth() / 2 + molecules[i].
                            getX(), 550 - 2, 4, 4);
```

```
        }
        int[] positiveSteps = new int[300];
        int[] negativeSteps = new int[300];
        for (int i = 0; i < molecules.length; i++) {
                int x = molecules[i].getX();
                if (x >= 0 && x < 300) {
                        positiveSteps[x]++;
                }
                else if (x <= 0 && x > -300) {
                        negativeSteps[Math.abs(x)]++;
                }
        }
        g2.setColor(Color.blue);
        for (int i = 0; i < positiveSteps.length; i++) {
                if (positiveSteps[i] > 0) {
                        g2.drawRect(this.getWidth() / 2 + i, 540
                            - positiveSteps[i], 1, positiveSteps[i
                            ] );
                }
        }
        for (int i = 1; i < negativeSteps.length; i++) {
                if (negativeSteps[i] > 0) {
                        g2.drawRect(this.getWidth() / 2 - i, 540
                            - positiveSteps[i], 1, positiveSteps[i
                            ] );
                }
        }

}

@SuppressWarnings("unchecked")
private void initComponents() {

        org.jdesktop.layout.GroupLayout layout = new org.jdesktop
            .layout.GroupLayout(this);
        this.setLayout(layout);
        layout.setHorizontalGroup(
```

```
                layout.createParallelGroup(org.jdesktop.layout.
                    GroupLayout.LEADING)
                .add(0, 400, Short.MAX_VALUE)
                );
                layout.setVerticalGroup(
                layout.createParallelGroup(org.jdesktop.layout.
                    GroupLayout.LEADING)
                .add(0, 300, Short.MAX_VALUE)
                );
        }
}
```

A.3. Molecule Class

```
package tr.edu.boun.cmpe.diffusion1D;

public class Molecule {

        private int X = 0;

        public int getX() {
                return X;
        }

        public void setX(int X) {
                this.X = X;
        }

        public void updateLocation() {
                if (SimulationParameters.RNG.nextBoolean()) {
                        X += SimulationParameters.MOLECULE_STEP_SIZE;
                }
                else {
                        X -= SimulationParameters.MOLECULE_STEP_SIZE;
                }
        }
}
```

A.4. PropagationEnvironment Class

```
package tr.edu.boun.cmpe.diffusion1D;

public class PropagationEnvironment {

        private Molecule[] molecules = null;

        public PropagationEnvironment() {
                molecules = new Molecule[SimulationParameters.
                    NUMBER_OF_MOLECULES];
                for (int i = 0; i < molecules.length; i++) {
                        molecules[i] = new Molecule();
                }
        }

        public void advanceTime() {
                for (int i = 0; i < molecules.length; i++) {
                        molecules[i].updateLocation();
                }
        }

        public Molecule[] getMolecules() {
                return molecules;
        }

        public void print() {
                for (int i = 0; i < molecules.length; i++) {
                        System.out.println("molecule [" + i + "]: " +
                            molecules[i].getX());
                }
        }
}
```

A.5. SimulationParameters Class

```
package tr.edu.boun.cmpe.diffusion1D;
```

```
import java.util.Random;

public class SimulationParameters {

        public static final int MOLECULE_STEP_SIZE = 1;
        public static final int NUMBER_OF_MOLECULES = 5000;
        public static final int SIMULATION_TIME = 1600;
        public static final int TIME_STEP = 1;
        public static final int SLEEP_TIME_FOR_GUI = 10;
        public static final Random RNG = new Random(System.
            currentTimeMillis());
}
```

APPENDIX B: SOURCE CODE for 2-D DIFFUSION

This chapter includes a sample implementation of Brownian motion for the simulation of a particle in two dimensions using Java programming language. A graphical user interface shows the path that the particle follows for the given time step.

B.1. Diffusion2D Class

```
package tr.edu.boun.cmpe.diffusion2D;

import javax.swing.JFrame;
import javax.swing.WindowConstants;

public class Diffusion2D {

        public static void main(String[] args) throws Exception {

                double currentTime = 0;
                PropagationEnvironment propagationEnvironment = new
                   PropagationEnvironment();

                JFrame frame = new JFrame("Diffusion 2D");
                frame.setSize(600, 600);
                frame.setDefaultCloseOperation(WindowConstants.
                   EXIT_ON_CLOSE);
                Diffusion2DGUI gui = new Diffusion2DGUI(
                   propagationEnvironment);
                frame.add(gui);
                frame.setVisible(true);
                while (currentTime < SimulationParameters.SIMULATION_TIME
                   ) {
                        propagationEnvironment.advanceTime();
                        currentTime += SimulationParameters.TIME_STEP;
                        gui.repaint();
```

```
                        Thread.currentThread().sleep(SimulationParameters
                            .SLEEP_TIME_FOR_GUI);
                }
                propagationEnvironment.print();
        }
}
```

B.2. Diffusion2DGUI Class

```
package tr.edu.boun.cmpe.diffusion2D;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Ellipse2D;
import java.awt.geom.Line2D;


public class Diffusion2DGUI extends javax.swing.JPanel {

        public Diffusion2DGUI(PropagationEnvironment pe) {
                propagationEnvironment = pe;
                initComponents();
        }

        public void paint(Graphics g) {
                Graphics2D g2 = (Graphics2D)g;
                double centerX = this.getWidth() / 2;
                double centerY = this.getHeight() / 2;
                Molecule molecule = propagationEnvironment.getMolecule();
                g2.setColor(Color.darkGray);
                g2.draw(new Line2D.Double(centerX + molecule.getPreviousX
                    () * 10,
                centerY + molecule.getPreviousY() * 10,
                centerX + molecule.getX() * 10,
                centerY + molecule.getY() * 10));
        }
```

```
        @SuppressWarnings("unchecked")
        private void initComponents() {

                org.jdesktop.layout.GroupLayout layout = new org.jdesktop
                    .layout.GroupLayout(this);
                this.setLayout(layout);
                layout.setHorizontalGroup(
                layout.createParallelGroup(org.jdesktop.layout.
                    GroupLayout.LEADING)
                .add(0, 400, Short.MAX_VALUE)
                );
                layout.setVerticalGroup(
                layout.createParallelGroup(org.jdesktop.layout.
                    GroupLayout.LEADING)
                .add(0, 300, Short.MAX_VALUE)
                );
        }
}
```

B.3. Molecule Class

```
package tr.edu.boun.cmpe.diffusion2D;

public class Molecule {

        private double X = 0;
        private double Y = 0;
        private double previousX = 0;
        private double previousY = 0;

        public double getX() {
                return X;
        }

        public void setX(double X) {
                this.X = X;
```

```
        }

        public double getY() {
                return Y;
        }
        public void setY(double Y) {
                this.Y = Y;
        }

        public double getPreviousX() {
                return previousX;
        }

        public void setPreviousX(double previousX) {
                this.previousX = previousX;
        }

        public double getPreviousY() {
                return previousY;
        }

        public void setPreviousY(double previousY) {
                this.previousY = previousY;
        }

        public void updateLocation() {
                setPreviousX(getX());
                setPreviousY(getY());

                setX(getX() + gaussian(0, SimulationParameters.SIGMA));
                setY(getY() + gaussian(0, SimulationParameters.SIGMA));
        }

        public static double gaussian(double mu, double sigma) {
                return mu + sigma * SimulationParameters.RNG.nextGaussian
                    ();
        }
}
```

B.4. PropagationEnvironment Class

```
package tr.edu.boun.cmpe.diffusion2D;

public class PropagationEnvironment {

        private Molecule molecule = null;

        public PropagationEnvironment() {
                molecule = new Molecule();
        }

        public void advanceTime() {
                molecule.updateLocation();
        }

        public Molecule getMolecule() {
                return molecule;
        }

        public void print() {
                System.out.println("molecule X: " + molecule.getX() + " Y
                    : " + molecule.getY());
        }
}
```

B.5. SimulationParameters Class

```
package tr.edu.boun.cmpe.diffusion2D;

import java.util.Random;

public class SimulationParameters {

        public static final double D = 79.4;
        public static final double TIME_STEP = 0.1;
        public static final double SIGMA = Math.sqrt(2 * D * TIME_STEP);
```

```
        public static final double SIMULATION_TIME = 10;
        public static final int SLEEP_TIME_FOR_GUI = 100;
        public static final Random RNG = new Random(System.
            currentTimeMillis());
}
```

APPENDIX C: SOURCE CODE for 3-D DIFFUSION

This chapter includes a sample implementation of Brownian motion for the simulation of particles in three dimensions using Java programming language. The program prints the probability of hit for the given time period.

C.1. Diffusion3D Class

```
package tr.edu.boun.cmpe.diffusion3D;
import javax.swing.JFrame;
import javax.swing.WindowConstants;

public class Diffusion3D {

        public static void main(String[] args) throws Exception {

                double currentTime = 0;
                PropagationEnvironment propagationEnvironment = new
                   PropagationEnvironment();
                while (currentTime < SimulationParameters.SIMULATION_TIME
                   ) {
                        propagationEnvironment.advanceTime();
                        currentTime += SimulationParameters.TIME_STEP;

                }
                propagationEnvironment.print();
        }
}
```

C.2. Molecule Class

```
package tr.edu.boun.cmpe.diffusion3D;

public class Molecule {
```

```
private double X = 0;
private double Y = 0;
private double Z = 0;

public double getX() {
        return X;
}

public void setX(double X) {
        this.X = X;
}

public double getY() {
        return Y;
}

public void setY(double Y) {
        this.Y = Y;
}

public double getZ() {
        return Z;
}

public void setZ(double Z) {
        this.Z = Z;
}

public void updateLocation() {
        setX(getX() + gaussian(0, SimulationParameters.SIGMA));
        setY(getY() + gaussian(0, SimulationParameters.SIGMA));
        setZ(getZ() + gaussian(0, SimulationParameters.SIGMA));
}

public static double gaussian(double mu, double sigma) {
        return mu + sigma * SimulationParameters.RNG.nextGaussian
            ();
}
```

```
}
```

C.3. PropagationEnvironment Class

```
package tr.edu.boun.cmpe.diffusion3D;

public class PropagationEnvironment {

        private Molecule[] molecules = null;
        private Receiver receiver = null;

        public PropagationEnvironment() {
                molecules = new Molecule[SimulationParameters.
                    NUMBER_OF_MOLECULES];
                for (int i = 0; i < molecules.length; i++) {
                        molecules[i] = new Molecule();
                }
                receiver = new Receiver();
        }

        public void advanceTime() {
                for (int i = 0; i < molecules.length; i++) {
                        if (molecules[i] != null) {
                                molecules[i].updateLocation();
                                if (receiver.isMoleculeReceived(molecules
                                    [i])) {
                                        molecules[i] = null;
                                }
                        }
                }

        }

        public Molecule[] getMolecules() {
                return molecules;
        }
```

```
        public void print() {
                System.out.println("Number of molecules sent: " +
                   SimulationParameters.NUMBER_OF_MOLECULES);
                System.out.println("Number of molecules received: " +
                   receiver.getNumberOfMoleculesReceived());
                System.out.println("Probability to capture (simulation):
                   " + receiver.getNumberOfMoleculesReceived() / (double)
                   SimulationParameters.NUMBER_OF_MOLECULES);
                System.out.println("Probability to capture (analytic): "
                   + receiver.getRadius() / receiver.getDistance(0, 0, 0,
                    receiver.getX(), receiver.getY(), receiver.getZ()));
        }

        public void printMolecules() {
                for (int i = 0; i < molecules.length; i++) {
                        if (molecules[i] != null) {
                                System.out.println("molecule [" + i + "]:
                                    X: " + molecules[i].getX() + " Y: " +
                                    molecules[i].getY() + " Z: " +
                                   molecules[i].getZ());
                        }
                }
        }
}
```

C.4. Receiver Class

```
package tr.edu.boun.cmpe.diffusion3D;

public class Receiver {

        private double X = 11;
        private double Y = 0;
        private double Z = 0;
        private double radius = 10;
        private int numberOfMoleculesReceived = 0;
```

```
public double getX() {
        return X;
}

public void setX(double X) {
        this.X = X;
}

public double getY() {
        return Y;
}

public void setY(double Y) {
        this.Y = Y;
}

public double getZ() {
        return Z;
}

public void setZ(double Z) {
        this.Z = Z;
}

public double getRadius() {
        return radius;
}

public void setRadius(double radius) {
        this.radius = radius;
}

public static double getDistance(double x1, double y1, double z1,
     double x2, double y2, double z2) {
        double diff1 = Math.pow(x1 - x2, 2);
        double diff2 = Math.pow(y1 - y2, 2);
        double diff3 = Math.pow(z1 - z2, 2);
```

```
                return Math.sqrt(diff1 + diff2 + diff3);
        }

        public boolean isMoleculeReceived(Molecule molecule) {
                double distance = getDistance(molecule.getX(),
                molecule.getY(),
                molecule.getZ(),
                getX(),
                getY(),
                getZ());
                if (distance <= radius) {
                        numberOfMoleculesReceived++;
                        return true;
                }
                else {
                        return false;
                }
        }

        public int getNumberOfMoleculesReceived() {
                return numberOfMoleculesReceived;
        }
}
```

C.5. SimulationParameters Class

```
package tr.edu.boun.cmpe.diffusion3D;
import java.util.Random;

public class SimulationParameters {

        public static final double D = 79.4;
        public static final double TIME_STEP = 0.001;
        public static final double SIGMA = Math.sqrt(2 * D * TIME_STEP);
        public static final double SIMULATION_TIME = 1000;
        public static final int NUMBER_OF_MOLECULES = 1000;
```

```
        public static final Random RNG = new Random(System.
            currentTimeMillis());

}
```

REFERENCES

1. Möhler, H., J. M. Fritschy, F. Crestani, T. Hensch, and U. Rudolph, "Specific GABAA Circuits in Brain Development and Therapy", *Biochemical Pharmacology*, Vol. 68, No. 8, pp. 1685–1690, 2004.

2. Akyildiz, I. F., F. Brunetti, and C. Blazquez, "Nanonetworks: A New Communication Paradigm", *Elsevier Computer Networks*, Vol. 52, No. 12, pp. 2260–2279, 2008.

3. Arora, K. V. and M. L. P. Tan, "Quantum Nanoelectronics: Challenges and Opportunities", *ICSE 2008 Proc*, 2008.

4. Jornet, J. M. and I. F. Akyildiz, "Graphene-based Nano-antennas for Electromagnetic Nanocommunications in the Terahertz Band", *Antennas and Propagation, 2010 Proceedings of the Fourth European Conference on, pp. 1*, EuCAP, 2010.

5. Freitas, R. A., *Nanomedicine, Volume I: Basic Capabilities*, Landes Bioscience Georgetown, TX, 1999.

6. Jornet, J. M. and I. F. Akyildiz, "Channel Modeling and Capacity Analysis for Electromagnetic Wireless Nanonetworks in the Terahertz Band", *Wireless Communications, IEEE Transactions on*, Vol. 10, No. 10, pp. 3211–3221, 2011.

7. Kuran, M. S., T. Tugcu, and B. O. Edis, "Calcium Signaling: Overview and Research Directions of a Molecular Communication Paradigm", *IEEE Wireless Communications*, Vol. 19, No. 5, pp. 20–27, 2012.

8. Nakano, T. and J.-Q. Liu, "Design and Analysis of Molecular Relay Channels: An Information Theoretic Approach", *NanoBioscience, IEEE Transactions on*, Vol. 9, No. 3, pp. 213–221, 2010.

9. Heren, A. C., M. S. Kuran, H. B. Yilmaz, and T. Tugcu, "Channel Capacity of Calcium Signalling based on Inter-cellular Calcium Waves in Astrocytes", *Communications Workshops (ICC), 2013 IEEE International Conference on*, pp. 792–797, IEEE, 2013.

10. Moore, M., E. A., T. Nakano, R. Egashira, T. Suda, A. Kayasuga, H. Kojima, H. Sakakibara, and O. K., "A Design of a Molecular Communication System for Nanomachines Using Molecular Motors", *Pervasive Computing and Com- munications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, IEEE, 2006.

11. Parcerisa Giné, L. and I. F. Akyildiz, "Molecular Communication Options for Long Range Nanonetworks", *Elsevier Computer Networks*, Vol. 53, No. 16, pp. 2753–2766, 2009.

12. Kim, N.-R. and C.-B. Chae, "Novel Modulation Techniques Using Isomers as Messenger Molecules for Nano Communication Networks via Diffusion", *IEEE Journal on Selected Areas in Communications*, Vol. 31, No. 12, pp. 847–856, 2013.

13. Acar, B., A. Akkaya, G. Genc, H. B. Yilmaz, M. S. Kuran, and T. Tugcu, *Understanding Communication via Diffusion: Simulation Design and Intricacies*, Book chapter in Modeling, Methodologies and Tools for Molecular and Nano-scale Communications, Springer, 2014.

14. Hiyama, S., Y. Moritani, T. Suda, R. Egashira, A. Enomoto, M. Moore, and T. Nakano, "Molecular Communication", *Proc. of NSTI Nanotechnology Conference (NANOTECH)*, pp. 391–394, 2005.

15. Nakano, T., T. Suda, T. Koujin, T. Haraguchi, and Y. Hiraoka, "Molecular Communication Through Gap Junction Channels: System Design, Experiments and Modeling", *Proc. of IEEE Bio-Inspired Models of Network, Information and Computing Systems (Bionetics)*, pp. 139–146, Budapest, Hungary, 2007.

16. Atakan, B. and O. B. Akan, "An Information Theoretical Approach for Molecular Communication", *Proc. of IEEE Bio-Inspired Models of Network, Information and Computing Systems (Bionetics)*, pp. 33–40, 2007.

17. Atakan, B. and O. B. Akan, "Deterministic Capacity of Information Flow in Molecular Nanonetworks", *Elsevier Nano Communication Networks*, Vol. 1, No. 1, pp. 31–42, 2010.

18. Moore, M.-J., T. Suda, and K. Oiwa, "Molecular Communication: Modeling Noise Effects on Information Rate", *IEEE Transactions on NanoBioscience*, Vol. 8, No. 2, pp. 169–180, 2009.

19. Garralda, N., I. Llatser, A. Cabellos-Aparicio, and M. Pierobon, "Simulation-based Evaluation of the Diffusion-based Physical Channel in Molecular Nanonetworks", *Proc. of IEEE International Conference on Computer Communications Workshops (INFOCOM)*, pp. 443–448, Shanghai, P. R. China, 2011.

20. Kuran, M. S., H. B. Yilmaz, T. Tugcu, and B. Özerman, "Energy Model for Communication via Diffusion in Nanonetworks", *Elsevier Nano Communication Networks*, Vol. 1, No. 2, pp. 86–95, 2010.

21. Kuran, M. S., H. B. Yilmaz, and T. Tugcu, "A Tunnel-based Approach for Signal Shaping in Molecular Communication", *Proc. of IEEE International Conference on Communications Workshops (ICC)*, pp. 776–781, 2013.

22. Kuran, M. S., H. B. Yilmaz, T. Tugcu, and I. F. Akyildiz, "Interference Effects on Modulation Techniques in Diffusion Based Nanonetworks", *Elsevier Nano Communication Networks*, Vol. 3, No. 1, pp. 65–73, 2012.

23. Nakano, T., M. J. Moore, F. Wei, A. V. Vasilakos, and J. Shuai, "Molecular Communication and Networking: Opportunities and Challenges", *IEEE Transactions on NanoBioscience*, Vol. 11, No. 2, pp. 135–148, 2012.

24. Farsad, N., B. H. Yilmaz, A. Eckford, C.-B. Chae, and W. Guo, "A Comprehensive Survey of Recent Advancements in Molecular Communication", *http://arxiv.org/pdf/1410.4258.pdf*, 2014.

25. Moore, M., A. Enomoto, T. Nakano, T. Suda, A. Kayasuga, H. Kojima, H. Sakakibara, and K. Oiwa, "Simulation of a Molecular Motor Based Communication Network", *Bio-Inspired Models of Network, Information and Computing Systems, 2006. 1st*, 2006.

26. Toth, A., D. Banky, and V. Grolmusz, "3-D Brownian Motion Simulator for High-Sensitivity Nanobiotechnological Applications.", *IEEE Transactions on NanoBioscience*, Vol. 10, No. 4, 2011.

27. Gul, E., B. Atakan, and O. B. Akan, "NanoNS: A Nanoscale Network simulator Framework for Molecular Communications", *Elsevier Nano Communication Networks*, Vol. 1, No. 2, pp. 138–156, 2010.

28. Garralda, N., I. Llatser, A. Cabellos-Aparicio, and M. Pierobon, "Simulation-based Evaluation of the Diffusion-based Physical Channel in Molecular Nanonetworks", *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 443 – 448, 2011.

29. Garralda, N., I. Llatser, A. Cabellos-Aparicio, E. Alarcon, and M. Pierobon, "Diffusion-based Physical Channel Identification in Molecular Nanonetworks", *Nano Communication Networks*, Vol. 2, No. 4, pp. 196 – 204, 2011.

30. Felicetti, L., M. Femminella, and G. Reali, "A simulation Tool for Biological Nanocommunication Systems", *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies - ISABEL '11*, 2011.

31. Felicetti, L., M. Femminella, and G. Reali, "A Simulation Tool for Nanoscale Biological Networks", *Nano Communication Networks*, Vol. 3, No. 1, pp. 2 – 18, 2012.

32. Felicetti, L., M. Femminella, and G. Reali, "Simulation of Molecular Signaling in Blood Vessels: Software Design and Application to Atherogenesis", *Nano Communication Networks*, Vol. 4, No. 3, pp. 98 – 119, 2013.

33. Felicetti, L., M. Femminella, G. Reali, P. Gresele, and M. Malvestiti, "Simulating an in Vitro Experiment on Nanoscale Communications by Using BiNS2", *Nano Communication Networks*, Vol. 4, No. 4, pp. 172 – 180, 2013.

34. Suzuki, J., H. Budiman, T. A. Carr, and J. H. DeBlois, "A Simulation Framework for Neuron-based Molecular Communication", *17th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, IES2013, pp. 103–113, 2013.

35. Yilmaz, H. B. and C.-B. Chae, "Simulation Study of Molecular Communication Systems with an Absorbing Receiver: Modulation and ISI Mitigation Techniques", *Simulation Modelling Practice and Theory*, Vol. 49, No. 0, pp. 136 – 150, 2014.

36. Nakano, T., Y. Okaie, and J.-Q. Liu, "Channel Model and Capacity Analysis of Molecular Communication with Brownian Motion", *Communications Letters, IEEE*, Vol. 16, No. 6, pp. 797 – 800, 2012.

37. Guo, W., S. Wang, A. Eckford, and J. Wu, "Reliable Communication Envelopes of Molecular Diffusion Channels", *Electronics Letters, IEEE*, Vol. 49, No. 19, pp. 1248 – 1249, 2013.

38. Srinivas, K. V., A. W. Eckford, and R. S. Adve, "Molecular Communication in Fluid Media: The additive Inverse Gaussian Noise Channel", *IEEE Transactions on Information Theory*, Vol. 58, No. 7, pp. 4678–4692, 2012.

39. Noel, A., K. C. Cheung, and R. Schober, "Using Dimensional Analysis to Assess Scalability and Accuracy in Molecular Communication", *Communications Workshops (ICC), 2013 IEEE International Conference on*, 2013.

40. Schulten, K. and I. Kosztin, *Lectures in Theoretical Biophysics*, University of Illi-

nois, 2000.

41. Yilmaz, H. B., A. C. Heren, T. Tugcu, and C.-B. Chae, "3-D Channel Characteristics for Molecular Communications with an Absorbing Receiver", *accepted to IEEE Communications Letters*, 2014.

42. Berg, H. C., *Random Walks in Biology*, Princeton University Press, New Jersey, 1993.

43. Redner, S., *A Guide to First-passage Processes*, Cambridge University Press, Cambridge, 2001.

44. Tyrrell, H. J. V. and K. R. Harris, *Diffusion in Liquids, A Theoretical and Experimental Study*, Butterworth Publishers, London, 1984.

45. Saxton, M. J., "Modeling 2D and 3D Diffusion", *Methods in Membrane Lipids*, pp. 295–321, Springer, New York City, 2007.

46. Heren, A. C., F. N. Kilicli, G. Genc, and T. Tugcu, "Effect of Messenger Molecule Decomposition in Communication via Diffusion", *to appear in 1st ACM International Conference on Nanoscale Computing and Communication*, ACM NANOCOM 2014.

47. Toth, A., D. Banky, and V. Grolmusz, "3-D Brownian Motion Simulator for High-Sensitivity Nanobiotechnological Applications.", *IEEE transactions on nanobioscience*, Vol. 10, No. 4, pp. 248 – 249, 2011.

48. Akkaya, A., G. Genc, and T. Tugcu, "HLA Based Architecture for Molecular Communication Simulation", *Elsevier Simulation Modelling Practice and Theory*, Vol. 42, No. 0, pp. 163 – 177, 2014.

49. "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)", *IEEE Std. 1516-2000*, 2000.

50. AbouRizk, S., Y. Mohamed, H. Taghaddos, F. Saba, and S. Hague, "Developing Complex Distributed Simulation for Industrial Plant", *Proceedings of the 2010 Winter Simulation Conference*, 2010.

51. Dahmann, J. S., "High Level Architecture for Simulation", *Distributed Interactive Simulation and Real Time Applications, 1997., First International Workshop on*, 1997.

52. Fujimoto, R. M., *Parallel and Distributed Simulation Systems*, John Wiley & Sons, Inc, Hoboken, 2000.

53. "The Portico Project", `http://www.porticoproject.org/`, [Accessed December 2014].

54. "The JGroups Project", `http://www.jgroups.org/`, [Accessed December 2014].

55. Bush, S. F., *Nanoscale Communication Networks*, Artech House, Boston, 2010.

56. Pierobon, M. and I. F. Akyildiz, "Noise Analysis in Ligand-binding Reception for Molecular Communication in Nanonetworks", *IEEE Transactions on Signal Processing*, Vol. 9, No. 59, pp. 4168–4182, 2011.

57. H. B. Yilmaz and C.-B. Chae, "Arrival Modeling for Molecular Communication via Diffusion", *Electronics Letters*, Vol. 23, No. 50, pp. 1667–1669, 2014.

58. Akkaya, A., T. Tugcu, B. H. Yilmaz, and C.-B. Chae, "Effect of Receptor Density and Size on Signal Reception in Molecular Communication via Diffusion with an Absorbing Receiver", *Communications Letters, IEEE*, Vol. PP, No. 99, pp. 1–1, 2014.

59. Gao, B., J. Fritschy, D. Benke, and H. Mohler, "Neuron-specific Expression of GABAA-receptor Subtypes: Differential Association of the Alpha 1-and Alpha 3-subunits with Serotonergic and GABAergic Neurons.", *Neuroscience*, Vol. 54,

No. 4, pp. 881–892, 1993.

60. Luddens, H. and E. R. Korpi, "Biological Function of GABAA/benzodiazepine Receptor Heterogeneity", *Journal of Psychiatric Research*, Vol. 29, No. 2, pp. 77 – 94, 1995.

61. Bureau, M. and R. W. Olsen, "Multiple Distinct Subunits of the Gamma-aminobutyric Acid-A Receptor Protein Show Different Ligand-binding Affinities.", *Molecular Pharmacology*, Vol. 37, No. 4, pp. 497–502, 1990.

62. Ruano, D., F. Araujo, A. Machado, A. L. de Blas, and J. Vitorica, "Molecular Characterization of Type I $GABA_A$ Receptor Complex from Rat Cerebral Cortex and Hippocampus", *Molecular Brain Research*, Vol. 25, No. 3, pp. 225–233, 1994.

63. Whiting, P. J., T. P. Bonnert, R. M. McKernan, S. Farrar, B. L. Bourdelles, R. P. Heavens, D. W. Smith, L. Hewson, M. R. Rigby, D. J. Sirinathsinghji, *et al.*, "Molecular and Functional Diversity of the Expanding GABA-A Receptor Gene Family", *Annals of the New York Academy of Sciences*, Vol. 868, No. 1, pp. 645–653, 1999.

64. Mody, I. and R. A. Pearce, "Diversity of Inhibitory Neurotransmission Through $GABA_A$ Receptors", *Trends in Neurosciences*, Vol. 27, No. 9, pp. 569–575, 2004.

65. Akkaya, A. and T. Tugcu, "Analog Molecular Communication in Nanonetworks", *Bio-Inspired Models of Networks, Information, and Computing Systems*, Vol. 103 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 171–182, Springer Berlin Heidelberg, 2012.