B. TECH. PROJECT REPORT

On

Bidirectional Relaying Techniques in Diffusion Based Molecular Communication

BY

Jonaq Niveer Sarma

DISCIPLINE OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

December 2016

Bidirectional Relaying Techniques in Diffusion Based Molecular Communication

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degrees

of
BACHELOR OF TECHNOLOGY
in
ELECTRICAL ENGINEERING

Submitted by:
Jonaq Niveer Sarma

Guided by:
Dr. Prabhat Kumar Upadhyay
Assistant Professor, Discipline of Electrical Engineering

INDIAN INSTITUTE OF TECHNOLOGY INDORE
December 2016

CANDIDATE'S DECLARATION

I hereby declare that the project entitled **"Bidirectional Relaying Techniques in Diffusion Based Molecular Communication"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in Electrical Engineering completed under the supervision of **Dr. Prabhat Kumar Upadhyay, Assistant Professor, Discipline of Electrical Engineering, IIT Indore** is an authentic work.

Further, I declare that I have not submitted this work for the award of any other degree elsewhere.

Signature and name of the student with date

CERTIFICATE by BTP Guide(s)

It is certified that the above statement made by the students is correct to the best of my knowledge.

Signature of BTP Guide(s) with dates and their designation

Preface

This report on "Bidirectional Relaying Techniques in Diffusion Based Molecular Communication" is prepared under the guidance of Dr. Prabhat Kumar Upadhyay.

Through this report, a detailed analysis of the two-time-slot bidirectional relaying techniques in molecular communication is provided. Explanations of the methodologies involved are presented in a lucid manner. Simulation results are included wherever possible.

Jonaq Niveer Sarma
B.Tech. IV Year
Discipline of Electrical Engineering
IIT Indore

Acknowledgements

I wish to thank Dr. Prabhat Kumar Upadhyay, Assistant Professor, IIT Indore for his kind support and valuable guidance. I wish to thank Satish K. Tiwari, Ph.D. Scholar, Department of Electrical Engineering, IIT Indore for his guidance all throughout the project.

It is their help and support, due to which we became able to complete the study and technical report.

Without their support this report would not have been possible.

Jonaq Niveer Sarma
B.Tech. IV Year
Discipline of Electrical Engineering
IIT Indore

Abstract

The field of molecular communication has inspired rigorous research into communication systems mimicking biological processes found throughout nature. A sound theoretical basis is being developed using interdisciplinary knowledge ranging from biochemistry to communication engineering. As the signal propagation time and signal decay with distance are high for diffusion based molecular communication, which can be treated as wireless communication in nanoscale, the use of relay nodes is desirable. Four-time-slot and three-time-slot bidirectional relay models can be analyzed using existing research. A two-time-slot bidirectional relay model can be developed using the concepts from the three-time-slot model.

This report develops a solid analytical framework to characterize the two-time-slot bidirectional relay scheme incorporating two separate types of molecules and fully absorbing receivers. Analytical bit error rate expressions are derived for the model. A particle tracking based simulator is employed to perform the related simulations.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction to Diffusion Based Molecular Communication

There has been rapid development in the fields of nanotechnology, biomedicine and biotechnology over the past few decades. Nanomachines capable of simple tasks have been developed. But to extend the capabilities of what simple nanomachines can achieve, communication among them is necessary. Nanonetworking and related terms have been defined in IEEE P1906.1 standard. Devices ranging in size from $0.1\ \mu m$ to $10\ \mu m$ in at least one dimension are termed nanomachines. Communication between such devices that is controlled or engineered by humans is termed nanonetworking [1].

So far we've seen two diverging approaches to the challenge of nanonetworking. One approach is to miniaturize the existing electromagnetic waves based communication, using novel materials such as graphene to act as antennas in the terahertz range. This approach enjoys a rich theoretical basis developed over a long period of time.

The other approach is to mimic the biological systems already present in nature. Theoretical framework regarding them is not as rich as for traditional electromagnetism based communication methods, but rich examples of fully functional communication systems among small structures using molecules as the messengers already exist in nature. A few notable examples are pheromone signaling predominant among animals, or the communication process in neuromuscular junctions, where acetylcholine molecules are used as messengers to trigger contraction in muscle cells. Communication using chemical signals as carriers of information is called molecular communication [2].

Molecular communication may be more favorable in certain situations where traditional communication methods face challenges, e.g., in networks of tunnels or in saline environments, electromagnetic waves suffer very high attenuation and path loss. Molecular communication may yield more reliable systems in such environments. In nanoscale systems, molecular communication systems enjoy the inherent biocompatibility and low energy requirement as compared to electromagnetic communication. Some potential messenger molecules are specific proteins or gold nanoparticles etc. [3]

1.1. Propagation Mechanisms

There are different mechanisms in nature juxtaposing ways to use molecules as messengers. A few schemes of propagation are mentioned below.

1.1.1. Free Diffusion

Information particles can propagate in a fluid medium using random thermal motion. There is no requirement of external energy source for diffusion based propagation. Every molecule follows Brownian motion and the behavior of a large number of molecules can be estimated by Fick's laws of diffusion. In nature, propagation using free diffusion can be seen widely, e.g., DNA binding molecules propagating over DNA segments to search for a binding site.

The motion of individual particles can be modelled as [3]

$$(x_i, y_i, z_i) = (x_{i-1},\ y_{i-1}, z_{i-1}) + (\Delta x_i,\ \Delta y_i, \Delta z_i) \tag{1}$$

where, (x_i, y_i, z_i) denotes the position of the molecule in the i^{th} time step.

For standard Brownian motion or Weiner process,

$$\Delta x_i \sim N(0,\ 2D\Delta t)$$

$$\Delta y_i \sim N(0,\ 2D\Delta t)$$

$$\Delta z_i \sim N(0,\ 2D\Delta t) \tag{2}$$

where, $N(\mu, \sigma^2)$ is a Normal random variable with mean, μ and variance, σ^2
Δt is the time step
and D is the diffusion coefficient

$$D = \begin{cases} \dfrac{k_B T}{6\pi\eta R_H} \ if\ Sm \gg S_{fluid} \\ \dfrac{k_B T}{4\pi\eta R_H} \ if\ Sm \approx S_{fluid} \end{cases} \tag{3}$$

where, $k_B = 1.38 \times 10^{-23}$J/K is the Boltzman constant

T is the absolute temperature

η is the dynamic viscosity of the fluid

R_H is the hydraulic radius of the molecule

S_m and S_{fluid} are the comparative sizes of the propagating molecule and the fluid molecules.

Molecule	D ($\mu m^2/s$)
DNA	0.81 to 53
Insulin	150
Sucrose	520
Glucose	600
Glycerol	930
Water	2100

Table 1: Diffusion coefficients of selected molecules in water at 25^oC. *Source:* [3]

1.1.2. Flow Assisted Propagation

Free diffusion can be very slow at transporting molecules over large distances. The motion can be assisted by introducing a flow in the environment. A flow from the transmitter to the receiver can greatly reduce the propagation time for messenger molecules. The motion can be simulated using a modified (2)

$$\Delta x_i \sim v_{x,i-1}(x_{i-1}, y_{i-1}, z_{i-1})\Delta t + N(0,\ 2D\Delta t)$$

$$\Delta y_i \sim v_{y,i-1}(x_{i-1}, y_{i-1}, z_{i-1})\Delta t + N(0,\ 2D\Delta t)$$

$$\Delta z_i \sim v_{z,i-1}(x_{i-1}, y_{i-1}, z_{i-1})\Delta t + N(0,\ 2D\Delta t) \quad (4)$$

where, $v_{x,i-1}(x_{i-1}, y_{i-1}, z_{i-1})$, $v_{y,i-1}(x_{i-1}, y_{i-1}, z_{i-1})$ and $v_{z,i-1}(x_{i-1}, y_{i-1}, z_{i-1})$ are the flow velocities in the x, y and z directions respectively [3].

1.1.3. Bacteria Assisted Propagation

In [4], a bacteria based communication system was proposed where the bacteria was loaded with information particles and attractant molecules were released by the receiver. The bacteria was guided by the concentration gradient in the medium.

1.1.4. Molecular Motors

Molecular motor proteins, e.g., Kinesin can be used to actively transport messenger molecules from transmitter to receiver [3]. Such systems already exist in human cells. These motor proteins quite literally walk over microtubule tracks present in the cytoplasm of cells.

1.1.5. Propagation Through Gap Junction

Intercellular calcium wave (ICW) is one of the intercellular communication systems in biological cells. When a cell is stimulated, it reacts by increasing the cytosolic Ca^{2+}concentration. This in turn stimulates a neighboring cell through the connections between neighboring cell membranes known as gap junctions. ICW can reach $200 - 350\ \mu m$ in one dimension at a speed of $15 - 27\ \mu m/s$ [5].

1.2. An Interesting Biological Example: Neurotransmitters

An example of diffusion based molecular communication is found in the synaptic cleft and neuromuscular junctions. As a response to an action potential reaching the synapse, it releases bursts of Acetylcholine (Ach) molecules to the cleft. The ACh molecules diffuse through the junction to trigger the post synaptic neuron or a contraction in the muscle tissue in case of neuromuscular junctions. The triggering molecules are absorbed in the process by the receptors (AChRs) in the target cell membrane. After the triggering is complete, Acetylcholinesterase (AChEs) molecules degrade the ACh molecules in the medium enabling further communication [6]. Some chemicals that inhibit AChE activity act as deadly chemical weapons (nerve gases) [7].

1.3. Parallels to Traditional Electromagnetic Communication

Although the molecular communication systems are significantly different from traditional communication methods using electromagnetic waves, there are parallels possible to carry over. For example, in molecular communication via diffusion, the modulation schemes used can be directly related with digital modulation schemes in communication using electromagnetic waves. In diffusion based molecular communication, the information can be modulated in the following properties:

1.3.1. Number of Particles

Information can be encoded in the number of particles transmitted, or in the concentration if the number of molecules is sufficiently large. This is analogous to amplitude shift keying in traditional EM waves based communication.

1.3.2. Type of Particles

Information can be encoded in the type or composition of the molecules released. This is analogous to frequency shift keying.

1.3.3. Release time

Information can be modulated in the time of release of the particles within a symbol period. This is analogous to phase shift keying in digital modulation for traditional electromagnetic communication.

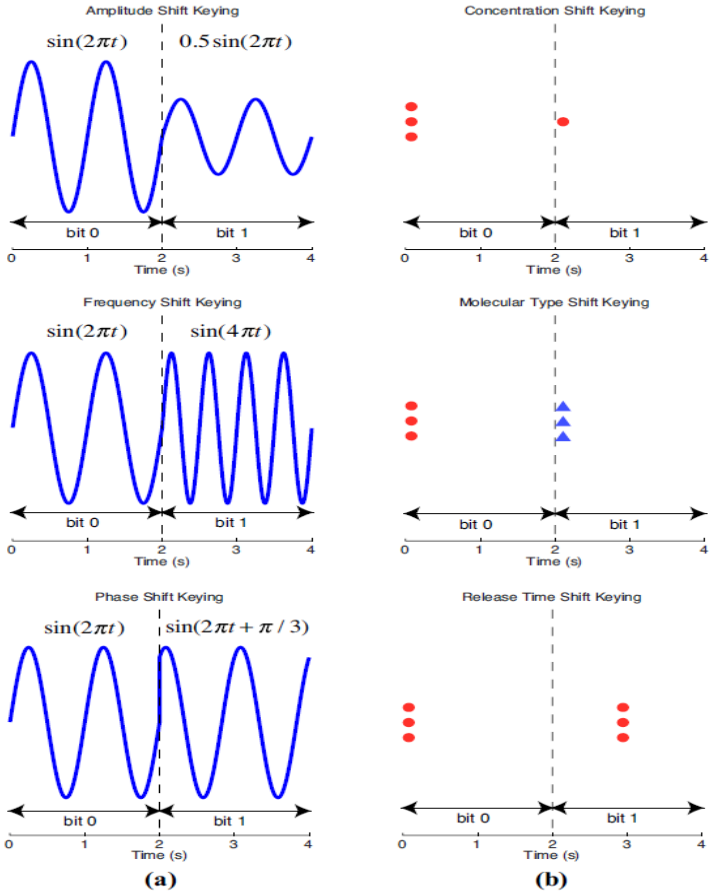

Figure 1: Modulation techniques in (a) traditional radio communication (b) molecular communication.

Source:[3]

Methods of analysis from information theory apply to molecular communication as well. Bit error rate and capacity calculations follow similar treatment as EM waves based communication. Thus, most of the analytical concepts carry over from the established communication engineering theory.

We constrain ourselves to systems using molecular communication via diffusion after this point. Also, our analysis is limited to physical layer design only. Higher levels can be approached using existing techniques.

Chapter 2

Point Source and Fully Absorbing Receiver

2.1. Motivation

As a stepping stone towards the bidirectional relaying model, we consider the simpler case of a point source of molecules and a fully absorbing receiver. More complex transmitters can be modelled using the point source model by assuming a superposition of point sources with infinitesimal number of molecules released. The channel model is discussed in the next section.

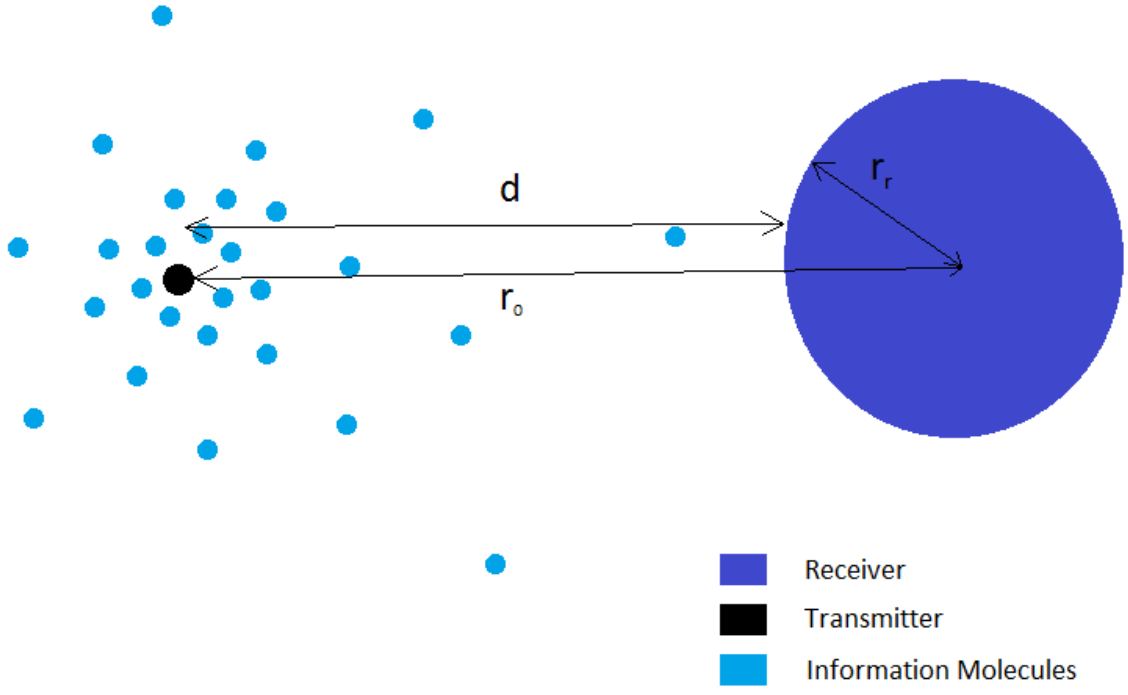

Figure 2: Channel Model

2.2. Channel Model

We assume a fully absorbing spherical receiver located at the origin with radius r_r and a point source of messenger molecules located at a distance r_0 from the origin, $r_0 > r_r$. The closest point on the receiver is at a distance d from the source. Therefore

$$d = r_0 - r_r$$

The source and the receiver are submerged in a fluid medium. The diffusion coefficient of the medium is given by

$$D = \begin{cases} \dfrac{k_B T}{6\pi\eta R_H} \; if \; Sm \gg S_{fluid} \\ \dfrac{k_B T}{4\pi\eta R_H} \; if \; Sm \approx S_{fluid} \end{cases} \tag{3}$$

where Sm and S_{fluid} are the relative sizes of the messenger molecules and the fluid molecules $k_B$is the Boltzman constant,

T is the temperature of the medium

η is the viscosity of the fluid medium

R_H is the hydraulic radius or Stoke's radius

D is assumed to remain constant for the communication period.

At $t = 0$, the source releases N number of molecules into the environment. The molecules propagate in the environment using diffusion dynamics, each molecule exhibiting standard Brownian motion. Molecules hitting the receiver surface are absorbed and removed from the environment. The receiver has inherent counting capabilities and can keep track of the number of molecules absorbed in the symbol duration T_s. The process can be estimated using the diffusion equation for large N.

2.3. Problem Formulation: Fick's Laws, Boundary Conditions

Fick's second law of diffusion gives

$$\frac{\partial C}{\partial t} = D\nabla^2 C \tag{5}$$

where, C is the concentration of the molecules

D is the diffusion coefficient

∇ is the Laplacian operator

For calculation purposes, we define, molecular distribution function $p(r,t)$ which is the probability of a single molecule released by the source being at a distance r from the origin. Adapting (5) to $p(r,t)$, we can write

$$\frac{\partial p(r,t)}{\partial t} = D\nabla^2 p(r,t) \tag{6}$$

The distribution function is assumed to be spherically symmetric. This assumption stems from the fact that the number of molecules hitting the absorbing surface $r = r_r$ will be same whether the molecules are released from a single point in space located at a distance $d = r_0 - r_r$ or from a spherical surface at distance $d = r_0 - r_r$. This gives an erroneous distribution function for other points in 3D space, but is accurate for calculations involving absorption at the spherical receiver. Thus, the initial condition is

$$p(r, t \to 0|r_0) = \frac{\delta(r - r_0)}{4\pi r_0} \tag{7}$$

We assume the molecule distribution to fade off at large distances

$$\lim_{r\to\infty} p(r,t|r_0) = 0 \tag{8}$$

For the final boundary condition, we create a boundary where collisions lead to absorption

$$D\frac{\partial p(r,t|r_0)}{\partial r} = \omega p(r,t|r_0) \text{ for } r \to r_r \tag{9}$$

where, w is the rate of reaction. For a fully absorbing receiver, $w \to \infty$ and

$$p(r_r,t|r_0) = 0 \tag{10}$$

also holds as a boundary condition.

2.4. Solution

Solving (6) with conditions (7), (8), (9) gives [8]

$$p(r,t|r_0) = \left(\frac{1}{4\pi r r_0}\right)\left(\frac{1}{\sqrt{4\pi Dt}}\right) \times \left(\exp\left[-\frac{(r-r_0)^2}{4Dt}\right] - \exp\left[-\frac{(r+r_0-2r_r)^2}{4Dt}\right]\right) - \left(\frac{1}{4\pi r r_0}\right)\left(\frac{wr_r+D}{Dr_r}\right) \times \exp\left[\left(\frac{wr_r+D}{Dr_r}\right)^2 Dt + \left(\frac{wr_r+D}{Dr_r}\right)(r+r_0-2r_r)\right] \times \operatorname{erfc}\left[\frac{wr_r+d}{Dr_r}\sqrt{Dt} + \frac{r+r_0-2r_r}{\sqrt{4Dt}}\right] \tag{11}$$

for $w \to \infty$, (11) becomes

$$p(r,t|r_0) = \left(\frac{1}{4\pi r r_0}\right)\left(\frac{1}{\sqrt{4\pi Dt}}\right)\left(e^{-\frac{(r-r_0)^2}{4Dt}} - e^{-\frac{(r+r_0-2r_r)^2}{4Dt}}\right) \tag{12}$$

Hitting rate of the molecules, $n_{hit}(R,t|r_0)$ to the receiver R is given by

$$n_{hit}(R,t|r_0) = \frac{r_r(r_0-r_r)}{r_0} \times \frac{1}{\sqrt{4\pi Dt^3}}\left[e^{-\frac{(r_0-r_r)^2}{4Dt}}\right] \tag{13}$$

Fraction of molecules absorbed till time t,

$$F_{hit}(R,t|r_0) = \int_0^t n_{hit}(R,t'|r_0)dt' = \frac{r_r}{r_0}\operatorname{erfc}\left(\frac{r_0 - r_r}{\sqrt{4Dt}}\right)$$

(14)

where

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = 1 - \left(\frac{2}{\sqrt{\pi}}\right)\int_0^x e^{-y^2}dy = \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-y^2}dy$$

(15)

Fraction of molecules arriving between time t and $t + \Delta t$ is given by

$$F_{hit}(R,t,t+\ \Delta t) = N_{hti}(R,t+\Delta t) - N_{hit}(R,t)$$

(16)

The responses are seen to exhibit a peak in absorbed molecules in both the theoretical approximation and the simulation. To calculate the peak time we set,

$$\frac{\partial n_{hit}}{\partial t} = 0$$

$$\therefore t = \frac{d^2}{6D}$$

(17)

This is in direct contrast with the case for EM waves. Notably, if we assume that the symbol time are in accordance with t_{peak}, the time taken for a message to be transmitted across a distance of $2d$ will take 4 times the time taken for the same message to travel across a distance of d.

Peak amplitude of n_{hit} can be formulated by substituting (17) in (14)

$$n_{peak} = \left(\frac{r_r}{r_0}\right) \times \sqrt{\frac{54}{\pi}} \times \left(\frac{D}{d^2}\right) \times e^{-\frac{3}{2}}$$

(18)

$$n_{peak} \sim \frac{1}{d^3}$$

A sample simulation is run in MATLAB using parameters from table 2. The code used is included in Appendix I

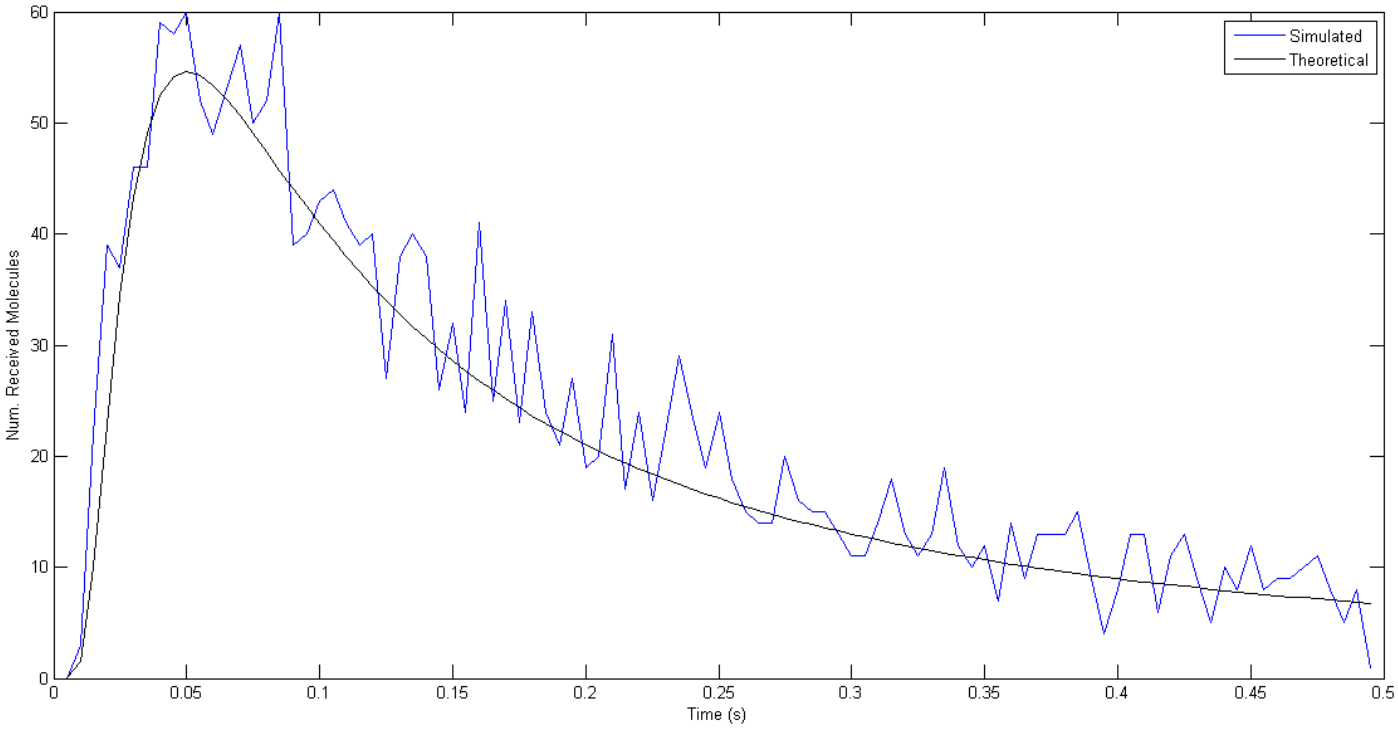

Figure 3: Sample simulation in MuCin using parameters from Table 2

Parameter	**Symbol**	**Value**
Distance of transmitter	r_0	$5\ \mu m$
Receiver radius	r_r	$2\ \mu m$
Diffusion coefficient	D	$20\ \mu m^2/s$
Total time of simulation	t	$0.5\ s$
Simulation step size	Δt	$5 \times 10^{-5} s$
Number of molecules released	N	10000

Table 2: Simulation parameters

Chapter 3

Two-Time-Slot Bidirectional Relaying

As estimated Chapter 2, diffusion based molecular communication suffers from high attenuation and use of relay nodes may be desirable. Literature have focused on both bidirectional relaying schemes [9], [10] and one-directional relaying [11], [12]. Here we focus on the bidirectional relaying scheme as proposed in [13].

3.1. Comparison to Other Bidirectional Relaying Schemes

There have been other methods of bidirectional relaying information in molecular communication literature. Some examples are stated below:

3.1.1. 4-Time-Slot Relaying

In a simpler relaying scheme, the two end nodes take one bit interval each to transmit their respective bits and the relay transmits them in the following two bit intervals to the opposite end nodes.

3.1.2. 3-Time-Slot Relaying

In this relaying scheme, during the two initial bit intervals, the end nodes transmit their information bits to the relay node. The relay node performs XOR operation on the detected bit and broadcasts the result to the end nodes in the last bit interval. The knowledge of the bit transmitted and the XOR result enables the end nodes to decode the information sent by the other relay node.

The above relaying schemes are feasible and are easier to analyze by treating each time slot individually. The bidirectional relaying model proposed in [13] increases the throughput of the system by reducing the number of time intervals required.

3.2. Channel Model

We assume a system with two nodes A and B whose information needs to be exchanged via a relay node R, situated between them, (Figure 4). Both the end nodes release their information bits using concentration shift keying in the same time slot (Multiple Access phase). The relay node counts the number of molecules hitting it and estimates whether both the end nodes transmitted the same bits or different. The relay node thus performs XOR operation on the detected bits and broadcasts the output of XOR operation in the next time slot (Broadcast phase). With the information of the transmitted bit from a node and the XOR output of the two bits from both nodes, the end nodes can decode the bit from the opposite end node. The end nodes transmit information using molecules of type p_1 and the relay node transmits using molecules of type p_2. This counters the issue of a node observing molecules transmitted by itself.

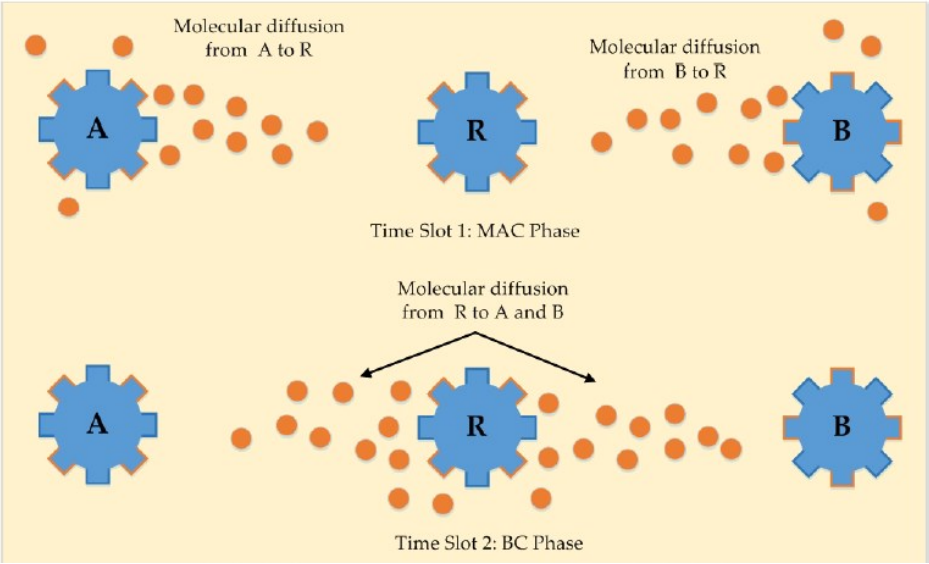

Figure 4: Channel Model. *Source* [13]

A	A XOR B	B (Decoded)
0	1	1
0	0	0
1	0	1
1	1	0

Table 3: Truth table used to decode the bits

3.3. Response

Each node being a receiver in addition to transmitters, it is reasonable to assume the nodes to be spherical. Thus, the transmission of molecules occurs from a spherical transmitter to a spherical receiver. We try to analyze systems with transmitters of finite dimensions by integrating (13). But as proposed in [14], for large distances between receiver and transmitter, the transmitter geometry affects the response very little, and the 'Point Transmitter Assumption' [14] gives satisfactory results. In addition, closed form analytical result is not obtained for channel impulse response for spherical transmitter and fully absorbing receiver and we have to rely on numerical approximations. As such, we refrain from modelling the transmitters as solid spheres and assume that all molecules to be released from the transmitter is initialized at the center of the transmitter. As such, probability of a molecule getting absorbed before time t is given by

$$F_{hit}(d_i, t) = \left(\frac{r_R}{r_0}\right) \operatorname{erfc}\left(\frac{r_0 - r_R}{\sqrt{4Dt}}\right) \tag{14}$$

For consistency in notation, we define $i \in \{A, B\}$. Let $s_i \in \{0, 1\}$ is the required bit to be sent to the relay from end node i in bit period t_b. If we consider on-off keying, the node emits N_i molecules if $s_i = 1$ and no molecules if $s_i = 0$.

The relay detects the sum of the transmitted bits,

$$s_R = s_A + s_B$$

$$s_R = \begin{cases} 2 \; if \; N_{RX_R} \geq \tau_2 \\ 1 \; if \; \tau_1 < N_{RX_R} < \tau_2 \\ 0 \; if \; N_{RX_R} < \tau_1 \end{cases} \tag{19}$$

where $\tau_1, \tau_2, \tau_1 < \tau_2$ are predetermined thresholds.

N_{RX_R}denotes the number of received molecules at relay node, R.

We make a simplifying assumption that all three nodes have the same radius r and distance from the center of a node to the surface of the nearest node is d. Therefore,

$$F_{hit}(d,t) = \left(\frac{r}{d-r}\right)\operatorname{erfc}\left(\frac{d}{\sqrt{4Dt}}\right)$$

(20)

Now assuming the probability of an individual molecule being received is given by $F_{hit}(d,t)$ (14), we can describe the number of molecules received in the transmitter using a Binomial random variable with N_{TX}trials and probability of success given by $F_{hit}(d,t)$.

$$N_{RX_i} \sim \mathcal{B}(s_i N_{TX}, F_{hit}(R_R, t_b|r_i))$$

(21)

Expected number of received molecules is

$$E(N_{RX}) = N_{TX}F_{hit}(d,t)$$

(22)

For very large N_{TX}, the number of molecules received approaches the deterministic model as developed in chapter 2, defined by Fick's laws of diffusion.

For probability of success, $F_{hit}(d,t)$ very small, the Binomial distribution can be estimated by a Poisson random variable.

On the other hand, for both $N_{TX}F_{hit}(d,t) > 5$ and $N_{TX}(1 - F_{hit}(d,t)) > 5$, N_{RX} can be estimated using a Normal random variable.

3.3.1. Multiple Access Phase

In the Multiple access phase, the number of molecules received by the relay node due to molecules transmitted by node i in the current bit period is given by

$$N_{Ri_c} \sim \mathcal{B}(s_i N_i, F_{hit}(R_R, t_b|r_i))$$

(23)

where, $\mathcal{B}(n,p)$ denotes a binomial random variable with number of trials n and probability of success p.

Similarly, contribution to the received signal from the past bit period is

$$N_{Ri_{p1}} \sim \mathcal{B}\big(s_i N_i, F_{hit}(\mathrm{R}_R, 3t_b|r_i)\big) - \mathcal{B}(s_i N_i, F_{hit}(\mathrm{R}_R, 2t_b|r_i))$$

(24)

Notice that we assume the relay node to keep absorbing molecules of type p_1 even during the transmitting period. Molecules of type $p_2$are not absorbed by the relay node and hence molecules released by R do not affect the received signal in R. The resultant received signal at R is

$$N_{Ri} = N_{Ri_c} + N_{RI_{p1}} + N_{Ri_{p2}} + \ \ldots$$

(25)

In order to make the above summation more approachable, we can approximate the Binomial random variable by Normal random variable.

$$N_{R_{i_c}} \sim \mathcal{N}(s_i N_i F_{hit}(R, t_b|r_i), s_i N_i F_{hit}(R_R, t_b|r_i)(1 - F_{hit}(R, t_b|r_i)))$$

(26)

where, $\mathcal{N}(\mu, \sigma^2)$ denotes a normal random variable with mean μ and variance σ^2

Similarly,

$$N_{Ri_{p_1}} \sim \mathcal{N}(s_i N_i F_{hit}(R, 3t_b|r_i), s_i N_i F_{hit}(R_R, 3t_b|r_i)\left(1 - F_{hit}\big(R, 3t_b|r_i\big)\right)\Big) - \mathcal{N}(s_i N_i F_{hit}(R, 2t_b|r_i), s_i N_i F_{hit}(R_R, 2t_b|r_i)\left(1 - F_{hit}\big(R, 2t_b|r_i\big)\right)\Big)$$

(27)

Assuming $N_A = N_B$

For ease of notation, let

$$\mu_k = N_A F_{hit}(R, kt_b|r_i)$$

$$\sigma_k^2 = N_A F_{hit}(R_R, kt_b|r_i)(1 - F_{hit}\big(R, kt_b|r_i\big))$$

(28)

Now, the probability of error during MAC phase, P_M relies on the s_i from previous time slots. As the most dominating factor in ISI is contributed by the bits transmitted in the immediate past [15], we can approximate

$$N_{Ri} \approx N_{Ri_C} + N_{Ri_{p1}} \tag{29}$$

As N_{Ri_C} and $N_{Ri_{p1}}$ depend on the transmitted bit, we can estimate the probability of error during MAC phase, P_M by Table 4. We assume bits to be equiprobable.

Assuming both bit values are equally likely, P_M can be estimated by,

$$\begin{aligned} P_M = \frac{1}{16}\Bigg[12 + 2Q\left(\frac{\tau_1 + \mu_2 - \mu_3}{\sqrt{\sigma_2^2 + \sigma_3^2}}\right) - 2Q\left(\frac{\tau_1 - \mu_1}{\sqrt{\sigma_1^2}}\right) + 2Q\left(\frac{\tau_2 - \mu_1}{\sqrt{\sigma_1^2}}\right) - 4Q\left(\frac{\tau_1 - \mu_1 + \mu_2 - \mu_3}{\sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}}\right) \\ + 4Q\left(\frac{\tau_2 - \mu_1 + \mu_2 - \mu_3}{\sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}}\right) + Q\left(\frac{\tau_1 + 2\mu_2 - 2\mu_3}{\sqrt{2\sigma_2^2 + 2\sigma_3^2}}\right) - 2Q\left(\frac{\tau_1 - \mu_1 + 2\mu_2 - 2\mu_3}{\sqrt{\sigma_1^2 + 2\sigma_2^2 + 2\sigma_3^2}}\right) \\ + 2Q\left(\frac{\tau_2 - \mu_1 + 2\mu_2 - 2\mu_3}{\sqrt{\sigma_1^2 + 2\sigma_2^2 + 2\sigma_3^2}}\right) - Q\left(\frac{\tau_2 - 2\mu_1}{\sqrt{2\sigma_1^2}}\right) - 2Q\left(\frac{\tau_2 - 2\mu_1 + \mu_2 - \mu_3}{\sqrt{2\sigma_1^2 + \sigma_2^2 + \sigma_3^2}}\right) \\ - Q\left(\frac{\tau_2 - 2\mu_1 + 2\mu_2 - 2\mu_3}{\sqrt{2\sigma_1^2 + 2\sigma_2^2 + 2\sigma_3^2}}\right)\Bigg] \end{aligned} \tag{30}$$

where $Q(x)$ is defined as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-u^2}\, du \tag{31}$$

Node A		Node B		Resultant Normal Distribution		Error occurs when N_{Ri}	P_M
Curr. bit	Past bit	Curr. Bit	Past bit	Mean	Variance		
0	0	0	0	0	0	$> \tau_1$	0
0	0	0	1	$\mu_3 - \mu_2$	$\sigma_2^2 + \sigma_3^2$	$> \tau_1$	$Q\left(\frac{\tau_1 + \mu_2 - \mu_3}{\sqrt{\sigma_2^2 + \sigma_3^2}}\right)$
0	0	1	0	μ_1	σ_1^2	$< \tau_1 or > \tau_2$	$1 - Q\left(\frac{\tau_1 - \mu_1}{\sqrt{\sigma_1^2}}\right) + Q\left(\frac{\tau_2 - \mu_1}{\sqrt{\sigma_1^2}}\right)$
0	0	1	1	$\mu_1 - \mu_2 + \mu_3$	$\sigma_1^2 + \sigma_2^2 + \sigma_3^2$	$< \tau_1 or > \tau_2$	$1 - Q\left(\frac{\tau_1 - \mu_1 + \mu_2 - \mu_3}{\sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}}\right) + Q\left(\frac{\tau_2 - \mu_1 + \mu_2 - \mu_3}{\sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}}\right)$
0	1	0	0	$\mu_3 - \mu_2$	$\sigma_2^2 + \sigma_3^2$	$> \tau_1$	$Q\left(\frac{\tau_1 + \mu_2 - \mu_3}{\sqrt{\sigma_2^2 + \sigma_3^2}}\right)$
0	1	0	1	$2\mu_3 - 2\mu_2$	$2\sigma_2^2 + 2\sigma_3^2$	$> \tau_1$	$Q\left(\frac{\tau_1 + 2\mu_2 - 2\mu_3}{\sqrt{2\sigma_2^2 + 2\sigma_3^2}}\right)$
0	1	1	0	$\mu_1 - \mu_2 + \mu_3$	$\sigma_1^2 + \sigma_2^2 + \sigma_3^2$	$< \tau_1 or > \tau_2$	$1 - Q\left(\frac{\tau_1 - \mu_1 + \mu_2 - \mu_3}{\sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}}\right) + Q\left(\frac{\tau_2 - \mu_1 + \mu_2 - \mu_3}{\sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}}\right)$
0	1	1	1	$\mu_1 - 2\mu_2 + 2\mu_3$	$\sigma_1^2 + 2\sigma_2^2 + 2\sigma_3^2$	$< \tau_1 or > \tau_2$	$1 - Q\left(\frac{\tau_1 - \mu_1 + 2\mu_2 - 2\mu_3}{\sqrt{\sigma_1^2 + 2\sigma_2^2 + 2\sigma_3^2}}\right) + Q\left(\frac{\tau_2 - \mu_1 + 2\mu_2 - 2\mu_3}{\sqrt{\sigma_1^2 + 2\sigma_2^2 + 2\sigma_3^2}}\right)$

1	0	0	0	μ_1	σ_1^2	$< \tau_1 or > \tau_2$	$1 - Q\left(\frac{\tau_1 - \mu_1}{\sqrt{\sigma_1^2}}\right) + Q\left(\frac{\tau_2 - \mu_1}{\sqrt{\sigma_1^2}}\right)$
1	0	0	1	$\mu_1 - \mu_2 + \mu_3$	$\sigma_1^2 + \sigma_2^2 + \sigma_3^2$	$< \tau_1 or > \tau_2$	$1 - Q\left(\frac{\tau_1 - \mu_1 + \mu_2 - \mu_3}{\sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}}\right) + Q\left(\frac{\tau_2 - \mu_1 + \mu_2 - \mu_3}{\sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}}\right)$
1	0	1	0	$2\mu_1$	$2\sigma_1^2$	$< \tau_2$	$1 - Q\left(\frac{\tau_2 - 2\mu_1}{\sqrt{2\sigma_1^2}}\right)$
1	0	1	1	$2\mu_1 - \mu_2 + \mu_3$	$2\sigma_1^2 + \sigma_2^2 + \sigma_3^2$	$< \tau_2$	$1 - Q\left(\frac{\tau_2 - 2\mu_1 + \mu_2 - \mu_3}{\sqrt{2\sigma_1^2 + \sigma_2^2 + \sigma_3^2}}\right)$
1	1	0	0	$\mu_1 - \mu_2 + \mu_3$	$\sigma_1^2 + \sigma_2^2 + \sigma_3^2$	$< \tau_1 or > \tau_2$	$1 - Q\left(\frac{\tau_1 - \mu_1 + \mu_2 - \mu_3}{\sqrt{\sigma_1^2 + \sigma_3^2 + \sigma_3^2}}\right) + Q\left(\frac{\tau_2 - \mu_1 + \mu_2 - \mu_3}{\sqrt{\sigma_1^2 + \sigma_3^2 + \sigma_3^2}}\right)$
1	1	0	1	$\mu_1 - 2\mu_2 + 2\mu_3$	$\sigma_1^2 + 2\sigma_2^2 + 2\sigma_3^2$	$< \tau_1 or > \tau_2$	$1 - Q\left(\frac{\tau_1 - \mu_1 + 2\mu_2 - 2\mu_3}{\sqrt{\sigma_1^2 + 2\sigma_2^2 + 2\sigma_3^2}}\right) + Q\left(\frac{\tau_2 - \mu_1 + 2\mu_2 - 2\mu_3}{\sqrt{\sigma_1^2 + 2\sigma_2^2 + 2\sigma_3^2}}\right)$
1	1	1	0	$2\mu_1 - \mu_2 + \mu_3$	$2\sigma_1^2 + \sigma_2^2 + \sigma_3^2$	$< \tau_2$	$1 - Q\left(\frac{\tau_2 - 2\mu_1 + \mu_2 - \mu_3}{\sqrt{2\sigma_1^2 + \sigma_2^2 + \sigma_3^2}}\right)$
1	1	1	1	$2\mu_1 - 2\mu_2 + 2\mu_3$	$2\sigma_1^2 + 2\sigma_2^2 + 2\sigma_3^2$	$< \tau_2$	$1 - Q\left(\frac{\tau_2 - 2\mu_1 + 2\mu_2 - 2\mu_3}{\sqrt{2\sigma_1^2 + 2\sigma_2^2 + 2\sigma_3^2}}\right)$

Table 4: Calculation for P_M

3.3.2. Broadcast Phase

During Broadcast phase, the end nodes have to decide whether the relay node R transmitted bit 0 or 1. Therefore only one threshold is required for decision

$$s_i = \begin{cases} 1 \; if \; N_{RX_i} \geq \tau \\ 0 \; if \; N_{RX_i} < \tau \end{cases}$$

(32)

Number of molecules received at node i from R, $i \in \{A, B\}$, in the current bit period is given by

$$N_{iR_c} \sim \mathcal{B}(s_R N_R, F_{hit}(R, t_b | r_R))$$

(33)

Number of molecules received at R residual from the past bit period is

$$N_{iR_{p1}} \sim \mathcal{B}(s_R N_R, F_{hit}(R, 3t_b | r_R)) - \mathcal{B}(s_R N_R, F_{hit}(R, 2t_b | r_R))$$

(34)

$$N_{R_i} = N_{R_{i_c}} + N_{R_{i_{p1}}} + N_{R_{i_{p2}}} + \ \ldots$$

(35)

Similar analysis as in MAC phase yields,

$$P_{B_A} = \frac{1}{4}\left[2 + Q\left(\frac{\tau + \mu_2 - \mu_3}{\sqrt{\sigma_2^2 + \sigma_3^2}}\right) - Q\left(\frac{\tau - \mu_1}{\sqrt{\sigma_1^2}}\right) - Q\left(\frac{\tau - \mu_1 + \mu_2 - \mu_3}{\sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}}\right)\right]$$

(36)

where

$$\mu_k = N_A F_{hit}(R_A, kt_b | r_R)$$

$$\sigma_k^2 = N_A F_{hit}(R_A, kt_b | r_R)(1 - F_{hit}(R_A, kt_b | r_R))$$

For P_M and P_{B_A} sufficiently low, i.e. $P_M, P_{B_A} \ll 1$, overall probability of error can be estimated as

$$P_e = P_M + P_{B_A}$$

(37)

3.4. ISI

Channel reuse capability in molecular communication is affected by the high inter symbol interferece without enzymatic degradation (Figure 5).

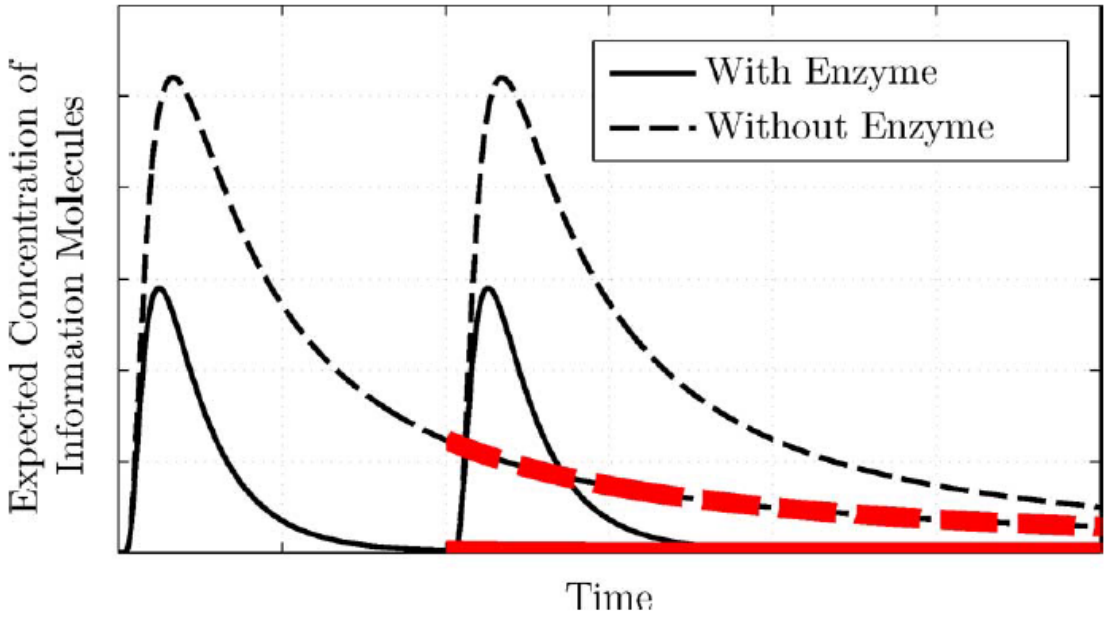

Figure 5: Effect of enzyme on n_{hit}. *Source:* [16]

As inspired in 1.2, presence of decaying enzymes in the medium can help greatly mitigate the inter symbol interference in molecular communication.

The degradation in a molecular channel can be described using the reaction [17]

$$E + S \leftrightarrow ES \rightarrow E + P \tag{38}$$

where E denotes the enzyme and S is the substrate, which in this case are the messenger molecules.

If we assume concentration of E, [E] to be very low and [S] to be very high, we can assume the reaction to be perfectly catalytic. Then

$$\frac{d[S]}{dt} = -\lambda[S] \Rightarrow [S] = [S]_0 e^{-\lambda t} \tag{39}$$

Taking Brownian motion and degradation as independent events [17],

P_r(molecule is received before it is degraded)

$$= F_{hit}(R, \lambda, t|r_0) = \int_0^t n_{hit}(R, t'|r_0) e^{-\lambda t'} dt'$$

$$= \frac{r_r}{r_0} e^{-\sqrt{\frac{\lambda}{t}}(r_0 - r_r)} - \frac{r_r}{2r_0} e^{-\sqrt{\frac{\lambda}{D}}(r_0 - r_r)} \times \left[\text{erf}\left(\frac{r_0 - r_r}{\sqrt{4Dt}} - \sqrt{\lambda t} \right) + e^{2\sqrt{\frac{\lambda}{D}}(r_0 - r_r)} \times \left[\text{erf}\left(\frac{r_0 - r_r}{\sqrt{4Dt}} + \sqrt{\lambda t} \right) - 1 \right] + 1 \right]$$

(40)

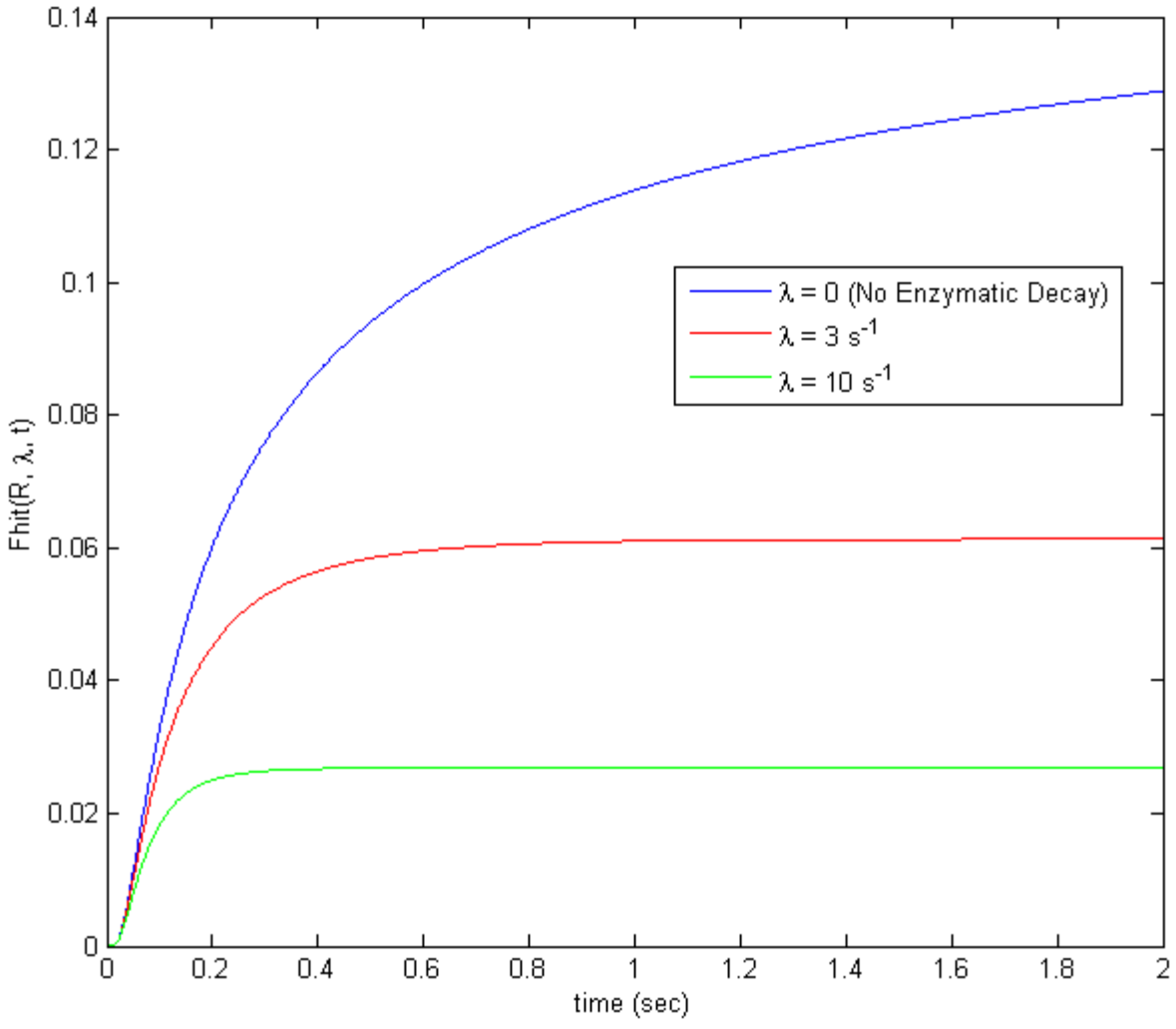

Figure 6: $F_{hit}(R, \lambda, t)$ as a function of time

F_{hit} for $\lambda > 0$ being lower than that without degradation, we can assume that the Poisson approximation is a more accurate model to model arrival rates of molecules [17]. But applying Poisson approximation in (24)

$$N_{R_{i_{p1}}} \sim \mathcal{P}\big(s_i N_i, F_{hit}(\mathrm{R}_R, 3t_b|r_i)\big) - \mathcal{P}(s_i N_i, F_{hit}(\mathrm{R}_R, 2t_b|r_i))$$

(41)

This gives rise to a Skellam Distribution and treating it is out of scope for our work.

Keeping with the Normal approximations yields similar formulations as (30) and (36) but,

$$\mu_k = N_A F_{hit}(R, \lambda, kt_b|r_i)$$

$$\sigma_k^2 = N_A F_{hit}(R, \lambda, kt_b|r_i)1 - F_{hit}\big(R, \lambda, kt_b|r_i\big)$$

(42)

Chapter 4

Simulation

4.1. Selection of the Simulation Framework

A comparison matrix of molecular communication simulators is given in Table 5. We opted for MuCin due to its open source nature and integration with MATLAB.

COMPARISON MATRIX OF MC SIMULATORS.

	dMCS	N3Sim	MUCIN	NanoNS	BINS	BNSim
Development language	Java	Java	MATLAB	NS-2, C++, Tcl	Java	Java
Parallelization	Yes	No	No	No	No	No
Open source	No	Yes	Yes	No	No	Yes
Propagation	Diffusion	Diffusion	Diffusion	Diffusion	Diffusion	Bacteria
Track each carrier	Yes	Yes	Yes	No	Yes	Yes
Reception	Absorption	Sampling	Absorption	Berg, Gillespie	Receptors	Receptors
Imperfect reception	No	No	Yes	No	Yes	No
Support unbounded medium	No	Yes[a]	Yes	Yes	Yes	No
Environment dimensions	3-D	2-D, 3-D[a]	1-D $\sim$ 3-D	3-D	3-D	3-D
Molecule interactions	No	No	No	No	Yes	No
Sending consecutive symbols	No	Yes[a]	Yes	No	No	Yes

[a] Possible only under specific conditions.

Table 5: Comparison matrix of simulators. *Source:* [3]

MuCin is an open source particle tracking based simulator developed and maintained by H. Birkan Yilmaz, and Chan-Byoung Chae. MuCin runs as a MATLAB package and provides tools ready to simulate systems with point source and spherical receiver, solid spherical source and spherical receiver, external molecule sources in addition to transmitter, concentration shift keying and molecule type shift keying as modulation techniques and variations in release patterns of the molecules. The simulator offers a particle tracking based simulator and a channel characteristic function based simulator, but we constrain ourselves to the particle tracking based simulator only due to the limited usefulness of the characteristic function based simulator.

4.2. Working principle

Particle tracking based simulators simulate diffusion process by tracking the Brownian motion of every constituent particle.

$$(x_i, y_i, z_i) = (x_{i-1},\ y_{i-1}, z_{i-1}) + (\Delta x_i,\ \Delta y_i, \Delta z_i) \tag{1}$$

$$\Delta x_i \sim N(0,\ 2D\Delta t)$$

$$\Delta y_i \sim N(0,\ 2D\Delta t)$$

$$\Delta z_i \sim N(0,\ 2D\Delta t) \tag{2}$$

$$D = \begin{cases} \dfrac{k_B T}{6\pi\eta R_H} \ if\ Sm \gg S_{fluid} \\ \dfrac{k_B T}{4\pi\eta R_H} \ if\ Sm \approx S_{fluid} \end{cases} \tag{3}$$

```
% Propagate the molecules via diffusion
mol_displace = normrnd (0, sigma, size(mol_position1,1), 3);
mol_position2 = mol_position1 + mol_displace;
```

Figure 7: The movement step in MuCin

4.3. Point source and circular receiver in 2D

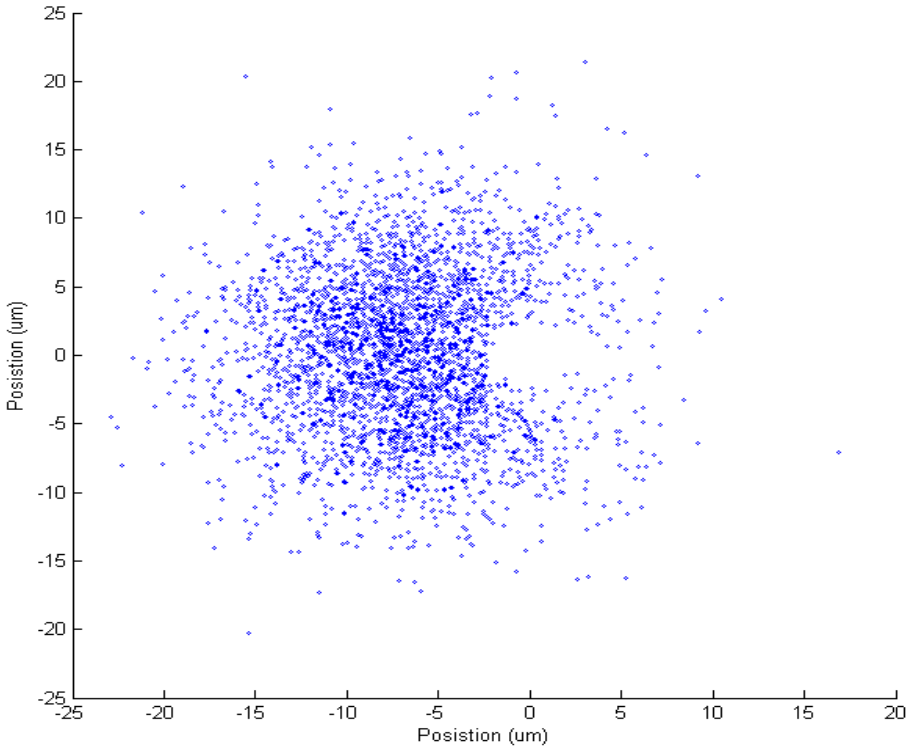

Figure 8: Scatter plot of molecule positions in a simulation in 2D plane. The receiver is fully absorbing. Simulation parameters are as given in Table 6.

Parameter	Value
Position of transmitter	$(-5\ \mu m, 0\ \mu m)$
Center of receiver	$(0\ \mu m, 0\ \mu m)$
Radius of the receiver	$2\ \mu m$
Diffusion Coefficient	$30\ \mu m^2/s$
Number of molecules	5000
Time into simulation	$0.5\ s$

Table 6: 2D simulation parameters

The simulation (Figure 7) shows the molecules scattered in 2D plane. This shows the concentration gradient established due to a point source and spherical receiver with impulse release on molecules.

4.4. Two-time-slot Bidirectional Relay Model

A sample simulation for the bidirectional relay model analyzed in chapter 3 is run using the parameters from Table 7. The MATLAB code used for the simulation is included in Appendix II.

The particle tracking based simulator is computationally very expensive. As new molecules are introduced in every bit interval, the total number of particles to track is quite high for the two-time-slot bidirectional relaying model discussed in Chapter 3. The simulator takes increasingly longer time to simulate subsequent uses of the channel. For the thousands of bits to simulate in order to gain insights into the relation of bit error rate to system parameters, simulation run time is too high. In the repeated run for 1000 random bits, the simulator gave 3 errors. Notably, the simulation run time was ~43 hours. Existing simulation framework is insufficient to measure correlation between BER and system parameters.

Parameter	**Symbol**	**Value**
Position of node A	A	$[-10\mu m, 0\mu m, 0\mu m]$
Position of node B	B	$[10\mu m, 0\mu m, 0\mu m]$
Position of node R	R	$[0\mu m, 0\mu m, 0\mu m]$
Radius of node A	r_A	$2\ \mu m$
Radius of node B	r_B	$2\ \mu m$
Radius of node R	r_R	$2\ \mu m$
Diffusion Coefficient	D	$50\mu m^2/sec$
Bit interval	t_b	$3\ sec$
Number of released molecules for bit 1	N_A, N_B, N_R	2000
Higher threshold for node R	τ_1	100
Lower threshold for node R	τ_2	350
Threshold for end nodes	τ	100
Simulation step size	Δt	$6\ ms$

Table 7: Simulation parameters for bidirectional relaying

Chapter 5

Results and Discussion

In order to gain insight into the expressions (30) and (36), Bit Error Rates (BER) as a function of individual system parameters are plotted (Figures 9 – 14). The default parameters are taken from Table 8 and individual parameters are varied.

System Parameter	Symbol	Value
Distance of receiver from transmitter	d	$10\ \mu m$
Diffusion coefficient of the medium	D	$300\ \mu m^2/s$
Number of molecules released for bit value of 1	N	3000
Receiver Radius	R	$2\ \mu m$
Rate of degradation	λ	$3\ s^{-1}$
Bit period	t_b	$1\ s$

Table 8: List of default values

Thresholds, τ_1, τ_2 and τ are calculated as

$$\tau_1 = \frac{\mu_1 - \mu_2 + \mu_3}{2}$$

$$\tau_2 = \frac{\mu_1 + \mu_2 + \mu_3}{2}$$

$$\tau = \frac{\mu_1}{2} \tag{43}$$

where, μ_1, μ_2 and μ_3 are defined in (28)

These values are chosen by intuition. Optimization is required to truly comment about the performance of the channel.

5.1. Effect of Distance on BER

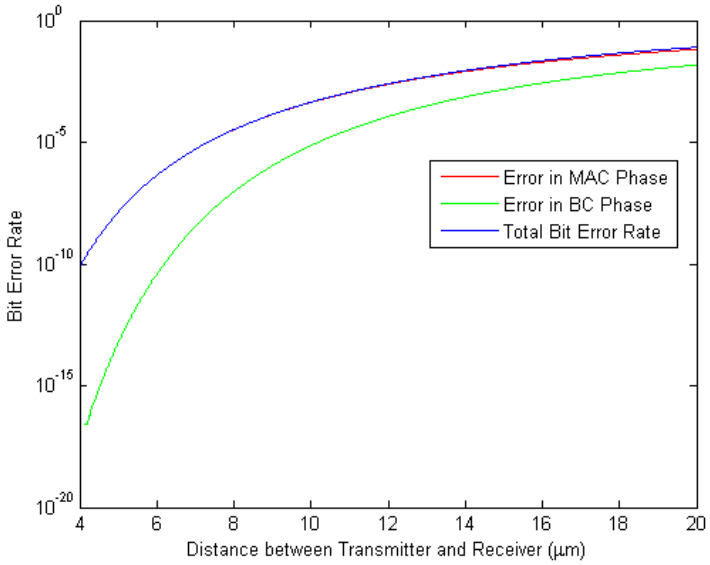

Figure 9: BER vs d

BER increases with the increase in distance between transmitter and receiver. Better capture probability for small distances improve BER performance.

5.2. Effect of Diffusion Coefficient on BER

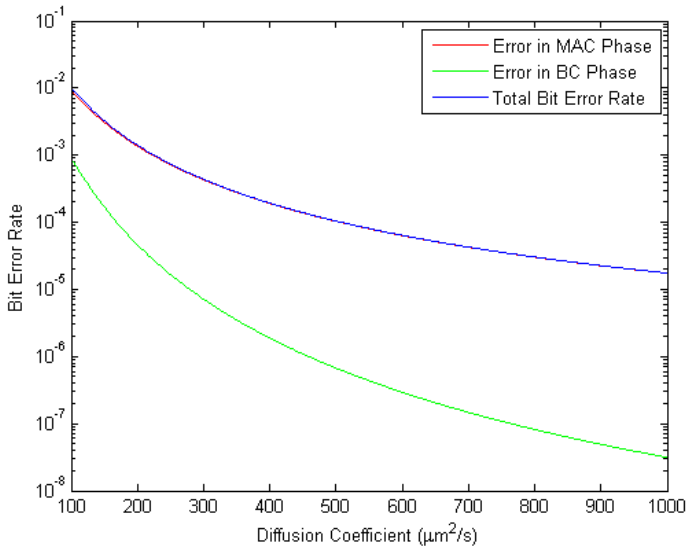

Figure 10: BER vs D

BER decreases with increase in the diffusion coefficient of the medium. Molecules have more mobility when diffusion coefficient increases and more capture probability is expected. Notably, diffusion coefficient varies with temperature and viscosity of the fluid medium.

5.3. Effect of the Number of Molecules Released on BER

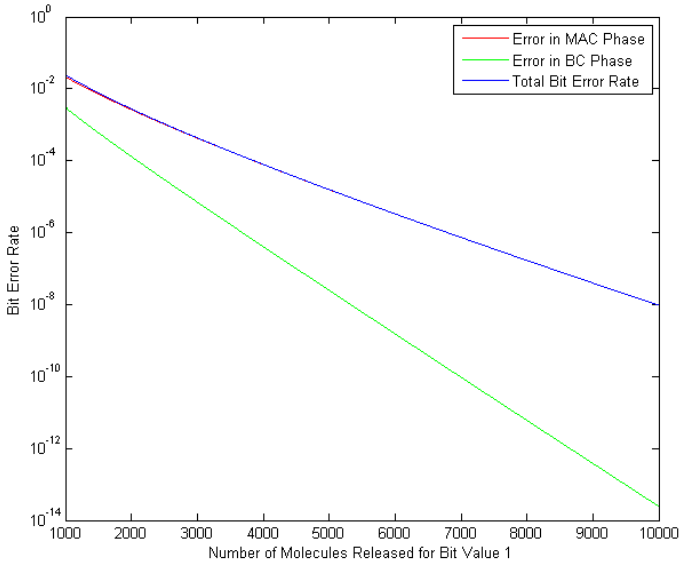

Figure 11: BER vs N

BER decreases with increase in the number of molecules released to represent bit 1. This is to be expected because the communication channel becomes deterministic for very large values of N, $N \to \infty$. Notably, error performance in broadcast phase is consistently better than that for multiple access phase. Therefore, number of molecules released in broadcast phase can be reduced without deteriorating BER.

5.4. Effect of Receiver Radius on BER

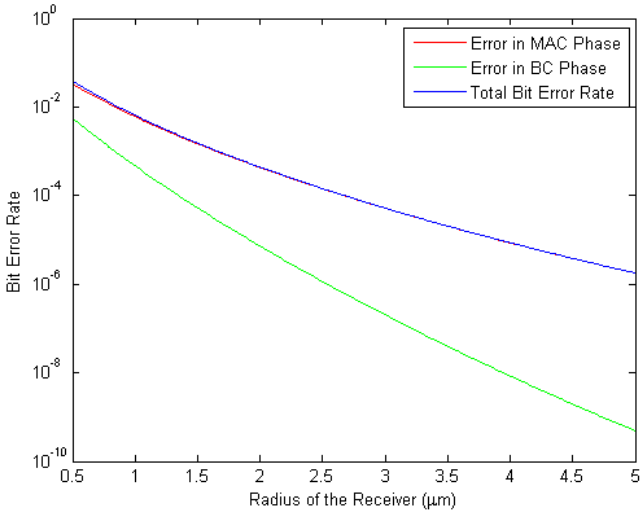

Figure 12: BER vs R

BER decreases with the increase in the receiver radius. Increasing receiver radius effectively increases capture probability and this aids in the reception.

5.5. Effect of Enzymatic Degradation on BER

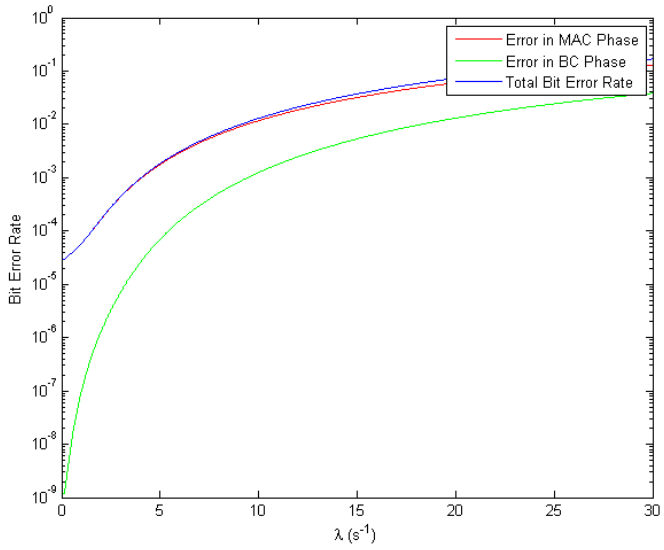

Figure 13: BER vs λ

Surprisingly, BER increases with increase in the rate of degradation. This can be attributed to the drastic decrease in capture probability in spite of decreased ISI. This case deserves future study, as initially the plot is convex and decision regions are not optimized in our work.

5.6. Effect of Bit Period on BER

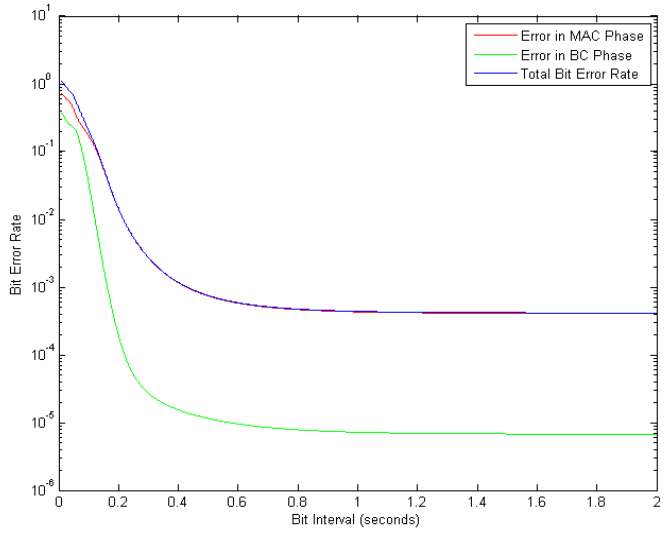

Figure 14: BER vs t_b

BER performance improves with increase in the bit interval, but it saturates as the capture probability saturates.

Chapter 6

Conclusion and Future work

We explored the existing research work in the field of molecular communication with specific concentration on the two time slot bidirectional relay model. The model was fully developed and the probabilistic model was applied to derive the expressions for bit error rate.

We acquainted ourselves with an open source simulation toolkit and derived own methods to simulate bidirectional relaying in MATLAB environment. The particle tracking based simulation technique was employed to simulate a bidirectional relay system but due to very high computational complexity, the correlation of bit error rate with different system parameters could not be obtained using the simulator.

Future work regarding the bidirectional relay model include

- Optimization of the decision regions to achieve minimum BER
- Further investigation of the effect of enzymes present in the medium
- Optimization of the iterative steps in the simulation framework to achieve faster outputs

As a rising field, molecular communication has inspired much interest from researchers. Advancements to the theoretical framework is continuing at a high pace. As such, new challenges are arising in the research as much as the older ones are tackled. We familiarized ourselves to the techniques involved in analyzing diffusion based molecular communication systems with the intention of contributing to the research in the field. There are rich opportunities for further research, exploring possibilities in molecular communication via diffusion.

References

1. T. Nakano, M. J. Moore, F.Wei, A. V. Vasilakos, and J. Shuai, “Molecular communication and networking: Opportunities and challenges,” *IEEE Trans. NanoBioscience*, vol. 11, no. 2, pp. 135–148, Jun. 2012.

2. T. Nakano, A. W. Eckford, and T. Haraguchi, *Molecular communication*, 1st ed. Cambridge University Press, 2013.

3. N. Farsad, H. B. Yilmaz, A. Eckford, C.-B. Chae, and W. Guo, “A comprehensive survey of recent advancements in molecular communication,” *IEEE Commun. Survey. Tutorials,* pp. 1–34, 2016.

4. L. C. Cobo and I. F. Akyildiz, “Bacteria-based communication in nanonetworks,” *Elsevier Nano Communication Network*, vol. 1, no. 4, pp. 244–256, Dec. 2010.

5. E. Scemes and C. Giaume, “Astrocyte calcium waves: what they are and what they do,” *Wiley Glia,* vol. 54, no. 7, pp. 716–725, Nov. 2006.

6. F. Mesiti and I. Balasingham, “Nanomachine-to-neuron communication interfaces for neuronal stimulation at nanoscale,” *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 695–704, Dec. 2013.

7. G. Bignami, N. Rosi´c, H. Michałek, M. Miloševi´c, and G. L. Gatti, “Behavioral toxicity of anticholinesterase agents: Methodological, neurochemical, and neuropsychological aspects,” in *Behavioral Toxicology*. New York, NY, USA: Springer, 1975, pp. 155–215.

8. H. B. Yilmaz, A. C. Heren, T. Tugcu, and C.-B. Chae, “Three dimensional channel characteristics for molecular communications with an absorbing receiver,” *IEEE Communication Letters*, vol. 18, no. 6, pp. 929–932, Jun. 2014.

9. M. F. Ghazani, G. Aminian, M. Mirmohseni, A. Gohari, and M. N. Kenari, “Physical layer network coding in molecular two-way relay networks,” in *Iran Workshop on Communication and Information Theory (IWCIT)*, May 2016, pp. 1–6.

10. B. D. Unluturk, D. Malak, and O. B. Akan, “Rate-delay tradeoff with network coding in molecular nanonetworks,” *IEEE Transactions on Nanotechnology,* vol. 12, no. 2, pp. 120–128, March 2013

11. A. Einolghozati, M. Sardari, and F. Fekri, “Decode and forward relaying in diffusion-based molecular communication between two populations of biological agents,” in *IEEE International Conference on Communications (ICC)*, June 2014, pp. 3975–3980.

12. T. Nakano and J. Q. Liu, “Design and analysis of molecular relay channels: An information theoretic approach,” *IEEE Transactions on NanoBioscience,* vol. 9, no. 3, pp. 213–221, Sept 2010.

13. Md. Noor-A-Rahim, MD Nashid Anjum and Guna Yong Liang, “Two-Time-Slot Bidirectional Relaying in Molecular Communication,” *arXiv:1609.01045*

14. Adam Noel, Dimitrios Makrakis and Abdelhakim Hafid, “Channel Impulse Responses in Diffusive Molecular Communication with Spherical Transmitters”, *arXiv:1604.04684*

15. M. Kuran, H. B. Yilmaz, T. Tugcu, and B. zerman, “Energy model for communication via diffusion in nanonetworks,” *Nano Communication Networks,* vol. 1, no. 2, pp. 86 – 95, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1878778910000219

16. Adam Noel, Karen C. Cheung, and Robert Schober, “Improving Receiver Performance of Diffusive Molecular Communication With Enzymes”, *IEEE Transactions on Nanobioscience,* vol. 13, no. 1, pp. 31-43 march 2014

17. Akif Cem Heren, H. Birkan Yilmaz, Chan-Byoung Chae, and Tuna Tugcu, ”Effect of Degradation in Molecular Communication: Impairment or Enhancement?”, *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications,* vol. 1, no. 2, june 2015 pp. 217-229

Appendix I

The MATLAB code below simulates a point source and spherical receiver system

```
%particle tracking based point source to spherical receiver simulator
function [nhit] = P2SwAbsorption(...
    txNode, ...      %Point source location in 3D
    rxNode, ...      %Center of receiver in 3D
    r_r, ...         %radius of receiver
    D, ...           %Diffusion coefficient
    nMolecules, ... %number of molecules to emit
    time, ...        %total time to simulate
    nStep )          %number of steps in the simulation

%calculate necessary parameters
displacement = txNode - rxNode;
d = sqrt( displacement(1)^2 ...
    + displacement(2)^2 ...
    + displacement(3)^2 ) ...
    - r_r;
tStep = time/nStep;
sigma = sqrt(2*D*tStep);

%prepare the particle tracking system,
AllMolecules = repmat(txNode, nMolecules, 1);
nhit = zeros(nStep, 1);

%move molecules, remove the absorbed molecules. count
for i = 1:nStep
   %notify about progress
   if mod(i, 20) == 0
       fprintf('%d steps in \n', i);
   end

   %move all existing molecules
   AllMolecules = AllMolecules + normrnd(0, sigma,
length(transpose(AllMolecules)), 3);

   %remove the absorbed molecules, count them.
   %Store in a temporary matrix during the calculations
   temp = zeros(1, 3);
   for j = 1:length(transpose(AllMolecules))
       if (...
               (AllMolecules(j, 1) - rxNode(1))^2 ...
               + (AllMolecules(j, 2) - rxNode(2))^2 ...
               + (AllMolecules(j, 3) - rxNode(3))^2 ...
               < r_r^2 )
           nhit(i, 1) = nhit(i, 1) + 1;
       else
           temp = [temp; AllMolecules(j, :)];
       end
```

```
    end
    temp = [temp(2:length(transpose(temp)), :)];
    AllMolecules = temp;
end

%plot
%reduce resolution for smoother plot
k = 10;      %merge factor
t = 0:tStep*k:time-tStep*k;
modified_nhit = zeros(0, 1);
for i=1:nStep/k
  temp = 0;
    for j = 1:k
       temp = temp + nhit((i-1)*k + j, 1);
    end
    modified_nhit = [modified_nhit; temp];
end
plot(t, modified_nhit);
end
```

Appendix II

The MATLAB code below simulates a two-time-slot bidirectional relay system

```
% simulation of bidirectional relay system
%Abits = [0 1 1 0];
%Bbits = [1 1 0 0];
Abits = round(rand(1000, 1));   %generation of random bits
Bbits = round(rand(1000, 1));

%environment
A = [-10 0 0];
B = [10 0 0];
R = [0 0 0];
Ra = 2;
Rb = 2;
Rr = 2;

D = 50;

%simulation steps for each t_b parameters
nStep = 500;
t_b = 3;
t_step = t_b/nStep;
sigma = sqrt(2*D*t_step);

%molecules released
Na = 2000;
Nb = 2000;
Nr = 2000;

tau1 = 100;
tau2 = 350;
tau = 100;

%set up the two types of molecules
AllMoleculesP1 = zeros(0,3);
AllMoleculesP2 = zeros(0,3);

decodedAbits = zeros(length(Abits), 1);
decodedBbits = zeros(length(Bbits), 1);

%simulation
for transmissionstep = 1:length(Abits)
    fprintf('simulating for bit %d \n', transmissionstep);
   %MAC phase
        %add new molecules
        AllMoleculesP1 = [AllMoleculesP1; ...
                repmat(A, Abits(transmissionstep)*Na, 1)];
        AllMoleculesP1 = [AllMoleculesP1; ...
```

```
            repmat(B, Bbits(transmissionstep)*Nb, 1)];

    %move all molecules. absorb all molecules. count only type P1, at
    %R
    nhitR = zeros(nStep, 1);
    for i = 1:nStep
        %notify about progress
            if mod(i, 20) == 0
                fprintf('%d/%d steps in MAC phase\n', i, nStep);
            end

    %move all existing molecules
    AllMoleculesP1 = AllMoleculesP1 ...
        + normrnd(0, sigma, length(transpose(AllMoleculesP1)), 3);
    AllMoleculesP2 = AllMoleculesP2 ...
        + normrnd(0, sigma, length(transpose(AllMoleculesP2)), 3);

    %remove the absorbed molecules, count only at realy node.
    %Store in a temporary matrix during the calculations
    temp = zeros(0, 3);
    for j = 1:length(transpose(AllMoleculesP1))
        if (...
                (AllMoleculesP1(j, 1) - R(1))^2 ...
                + (AllMoleculesP1(j, 2) - R(2))^2 ...
                + (AllMoleculesP1(j, 3) - R(3))^2 ...
                < Rr^2 )
            nhitR(i, 1) = nhitR(i, 1) + 1;
        else
            temp = [temp; AllMoleculesP1(j, :)];
        end
    end
    %temp = [temp(2:length(transpose(temp)), :)];
    AllMoleculesP1 = temp;

    temp = zeros(0, 3);
    for j = 1:length(transpose(AllMoleculesP2))
        if (...
                (AllMoleculesP2(j, 1) - A(1))^2 ...
                + (AllMoleculesP2(j, 2) - A(2))^2 ...
                + (AllMoleculesP2(j, 3) - A(3))^2 ...
                < Ra^2 )
            %don't do anything
        elseif(...
                (AllMoleculesP2(j, 1) - B(1))^2 ...
                + (AllMoleculesP2(j, 2) - B(2))^2 ...
                + (AllMoleculesP2(j, 3) - B(3))^2 ...
                < Rb^2 )
            %don't do anything
        else
            temp = [temp; AllMoleculesP2(j, :)];
        end
    end
    %temp = [temp(2:length(transpose(temp)), :)];
    AllMoleculesP2 = temp;

end
```

```
        %decode information
        Rbit = 1;
        %sum all nhitR. decide on Rbit
        if ( sum(nhitR) > tau2)
            Rbit = 0;
        elseif ( sum(nhitR(:)) < tau1)
            Rbit = 0;
        end

        %fprintf(1, 'read total of %d p1 molecules. XOR is %d',...
sum(nhitR),Rbit);

   %broadcast phase
        %add new molecules
        AllMoleculesP2 = [AllMoleculesP2; repmat(R, Rbit*Nr, 1)];

        %move all molecules. absorb all molecules. count only type P2, at
        %A and B
        nhitA = zeros(nStep, 1);
        nhitB = zeros(nStep, 1);

        for i = 1:nStep
            %notify about progress
                if mod(i, 20) == 0
                    fprintf('%d/%d steps in broadcast phase\n', i, nStep);
                end

        %move all existing molecules
        AllMoleculesP1 = AllMoleculesP1 ...
            + normrnd(0, sigma, length(transpose(AllMoleculesP1)), 3);
        AllMoleculesP2 = AllMoleculesP2 ...
            + normrnd(0, sigma, length(transpose(AllMoleculesP2)), 3);

        %remove the absorbed molecules, count only at end nodes.
        %Store in a temporary matrix during the calculations
        temp = zeros(0, 3);
        for j = 1:length(transpose(AllMoleculesP1))
            if (...
                    (AllMoleculesP1(j, 1) - R(1))^2 ...
                    + (AllMoleculesP1(j, 2) - R(2))^2 ...
                    + (AllMoleculesP1(j, 3) - R(3))^2 ...
                    < Rr^2 )
                %do nothing
            else
                temp = [temp; AllMoleculesP1(j, :)];
            end
        end
        %temp = [temp(2:length(transpose(temp)), :)];
        AllMoleculesP1 = temp;

        temp = zeros(0, 3);
        for j = 1:length(transpose(AllMoleculesP2))
            if (...
                    (AllMoleculesP2(j, 1) - A(1))^2 ...
```

```
                + (AllMoleculesP2(j, 2) - A(2))^2 ...
                + (AllMoleculesP2(j, 3) - A(3))^2 ...
                < Ra^2 )
            nhitA(i) = nhitA(i) + 1;
        elseif(...
                (AllMoleculesP2(j, 1) - B(1))^2 ...
                + (AllMoleculesP2(j, 2) - B(2))^2 ...
                + (AllMoleculesP2(j, 3) - B(3))^2 ...
                < Rb^2 )
            nhitB(i) = nhitB(i) + 1;
        else
            temp = [temp; AllMoleculesP2(j, :)];
        end
    end
    %temp = [temp(2:length(transpose(temp)), :)];
    AllMoleculesP2 = temp;

    end

    %decode information
    XORa = 0;
    XORb = 0;

    if sum(nhitA) > tau
        XORa = 1;
    end
    if sum(nhitB) > tau
        XORb = 1;
    end

    %fprintf('A received %d molecules\nB received %d molecules\n',
sum(nhitA), sum(nhitB));

    %decode B bits at node A
    %perform xor operation
    if (Abits(transmissionstep) == 0 && XORa == 0)
        decodedBbits(transmissionstep) = 0;
    elseif (Abits(transmissionstep) == 0 && XORa == 1)
        decodedBbits(transmissionstep) = 1;
    elseif (Abits(transmissionstep) == 1 && XORa == 0)
        decodedBbits(transmissionstep) = 1;
    elseif (Abits(transmissionstep) == 1 && XORa == 1)
        decodedBbits(transmissionstep) = 0;
    end

    %decode A bits at node B
    %perform xor operation
    if (Bbits(transmissionstep) == 0 && XORb == 0)
        decodedAbits(transmissionstep) = 0;
    elseif (Bbits(transmissionstep) == 0 && XORb == 1)
        decodedAbits(transmissionstep) = 1;
    elseif (Bbits(transmissionstep) == 1 && XORb == 0)
        decodedAbits(transmissionstep) = 1;
    elseif (Bbits(transmissionstep) == 1 && XORb == 1)
```

```
            decodedAbits(transmissionstep) = 0;
        end

        %fprintf('decoded A = %d, B = %d', decodedAbits(transmissionstep),
decodedBbits(transmissionstep));
end

errors = 0;
for i = 1:length(transpose(Abits))
    if (decodedAbits(i) - Abits(i)) ~= 0
        errors = errors+1;
    elseif (decodedBbits(i) - Bbits(i)) ~= 0
        errors = errors+1;
    end
end
   errors
```