Understanding Communication via Diffusion: Simulation Design and Intricacies

Bilal Acar, Ali Akkaya, Gaye Genc, H. Birkan Yilmaz, M. Şükrü Kuran and Tuna Tugcu

Abstract Understanding Communication via Diffusion (CvD) is key to molecular communications research since it dominates the movement at the nano-scale. The researcher needs to properly understand the random diffusion of the molecules for the analysis of a molecular communication system. This chapter aims explaining the dynamics of diffusion from a communication engineer's perspective as well as providing useful hints for an effective simulation design by discussing some key intricacies. The chapter starts with a brief survey of simulators for molecular communications, followed by the basics of the simulation of Brownian motion and CvD. Several intricacies are addressed to help the researcher in simulation design, such as the number of replications required in terms of movement and bit sequence. We utilize this information further by discussing the design of more complex CvD systems such as tunnel-based approach that utilizes destroyer molecules and distributed simulator design based on HLA. Introduction of more complex CvD systems provides significant improvements in data rate and communications in general, bridging the gap between human-scale and nano-scale systems and enabling nanonetworking as a viable technology.

B. Acar · A. Akkaya · G. Genc · H.B. Yilmaz · T. Tugcu (✉)
NETLAB, Department of Computer Engineering, Bogazici University,
34342 Istanbul, Turkey
e-mail: tugcu@boun.edu.tr

B. Acar
e-mail: bilal.acar.a@gmail.com

A. Akkaya
e-mail: ali.akkaya@boun.edu.tr

G. Genc
e-mail: gaye.genc@boun.edu.tr

H.B. Yilmaz
School of Integrated Technology, Yonsei University, Seoul, South Korea
e-mail: birkan.yilmaz@boun.edu.tr

M. Şükrü Kuran
Abdullah Gul University, Kayseri, Turkey
e-mail: sukru.kuran@agu.edu.tr

J. Suzuki et al. (eds.), *Modeling, Methodologies and Tools for Molecular and Nano-scale Communications*, Modeling and Optimization in Science and Technologies 9, DOI 10.1007/978-3-319-50688-3_7

1 Introduction

Communication via Diffusion (CvD) is one of the key communication mechanisms in the context of Nanonetworks that is proposed by the information and communication technology literature. As the most basic definition, CvD deals with communication systems between transmitter-receiver couples; exchanging information via a shared medium using molecules that represent the data. The information is usually regarded as being encoded onto the quantity of the molecules. Thus, the receiver decodes the information based on the number of molecules it receives within a predefined time period (symbol duration). Regardless of the specific subsystem that is being considered, the medium is assumed to be uncontrolled, or partially controlled, and the molecules propagate through it following the prevalent diffusion dynamics in this miniscule scale. These dynamics define a very unique physical layer unlike the conventional wired and wireless communication media, which brings new challenges specific for this communication system.

Diffusion, as a movement model for small particles, has been studied extensively in the 19th and early 20th century by scientists like Thomas Graham and Adolf Fick. Diffusion focuses on capturing the general behavior 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 behavior is actually the macroscopic result of some basic movement that is conducted in a microscopic scale, called the Brownian motion. In other words, the movement of individual small particles can be modelled by Brownian motion dynamics.

According to the current literature on diffusion, several system parameters of diffusing particles in several topologies can be evaluated mathematically such as the average arrival time of diffusing particles from a single point source to a given spherical boundary in a 3-D space or the upper bound on the probability of hitting to a spherical body when particles are released again from a given single point source. Nonetheless, when the topology of the system becomes more complex the analytical solution to these parameters become harder to evaluate. For example, in a slightly more complex setting consisting of two spherical bodies, the upper bound of particles hitting to a sphere when released from another sphere is not an easy task and has not been solved analytically yet. On the other hand, in such complex topologies, these system parameters can be evaluated via simulation using Brownian motion dynamics in a short period of time. Therefore, simulation is a crucial method for analyzing the potential performance metrics of a given CvD system.

The research activities on molecular communication utilizes simulations to verify and analyze the performance of proposed models. Due to the different channel characteristics of the fluid environment and the carrier wave properties, current simulation tools can not be used directly for nanonetworking. Simulation of molecular communication requires modeling the new communication paradigm that comprises different options for transmission, propagation, and reception. It should consider possible architectural design options and performance evaluation of molecular communication networks. Since molecular communication involves the modeling of large

number of nano-scale objects, scalability of the simulation tool is another important concern. 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. Another challenge for research groups is to generate a simulation execution plan. 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.

In the literature, besides the use of simulation as a research tool, there are several works that have been performed specifically on nano-scale simulation design. In [17], the need for simulation is mentioned and simulation requirements of molecular motor based communication network is briefly defined. In [20], a simulator for 3-D Brownian motion is proposed. The simulator is capable of modeling nano particles under various configurable circumstances to simulate molecule diffusion and reception. The novelty of the proposed model is a dual time step approach to cope with the run time complexity of 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. In [12], the authors introduce a C++ and Tcl based simulation framework developed on top of the commonly used NS-2 discrete event simulator targeted for networking 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 [2] 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 [15]. In [9], a simulation platform for modeling information exchange at the 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. The model is defined to be generic and can be used for different communication options. A case study is used to demonstrate the features of the simulation tool. In [4], a distributed architecture for molecular communication is proposed. The architecture is built on top of High Level Architecture (HLA), and provides interoperable, re-usable, and scalable design options for simulation of molecullar communication paradigm.

Both the simulator design, and the simulation execution plan are important steps of molecular communication research process. 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, and affect the evaluation of alternative options for the proposed model. Flexibility, re-usability, interoperability, and scalability should be considered during simulator

design process. Based on the research project needs, 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 results. This chapter aims to guide the reader on the simulator design issues and simulation execution plan. This will help the reader to better plan and execute the simulation step of molecular communication research activities, which in general dominates the overall research plan.

2 Simulation Design of CvD

2.1 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: 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

$$J = -D\frac{\partial C(x,t)}{\partial x} \tag{1}$$

where x is the position, $C(x,t)$ is the concentration at that point, and D is the diffusion constant. 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

$$(J(x + \Delta x) - J(x))\Delta t = -\Delta C \Delta x. \tag{2}$$

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

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2} \tag{3}$$

Equation 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

$$\frac{\partial C(x,t)}{\partial t} = D\nabla^2 C(x,t) \tag{4}$$

where ∇^2 is the Laplacian operator.

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

$$C(x,t) = \frac{1}{(4\pi Dt)^{m/2}} e^{-||x||^2/4Dt} \tag{5}$$

where m and D are the dimension of the environment and the diffusion coefficient, respectively [18]. The value of D depends on the temperature of the environment, viscosity of the fluid, and the Stokes' radius of the molecule [21].

The microscopic theory of diffusion is utilized for simulating the motion of diffusing particles. Brownian motion, which can be seen as a discrete case of the diffusion process, is simulated by the help of a *good* random number generator. (Based on our experience, we strongly encourage the use of a random number generator derived from Mersenne Twister). For the simulation process, we do not consider the collisions between particles for the sake of simplicity. 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

$$\Delta X \sim \mathcal{N}(0, \sigma^2) \tag{6}$$

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 [6]. As an alternative option for simulating the Brownian motion, one may select the direction randomly and move same amount. Both schemes are equivalent when the 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.

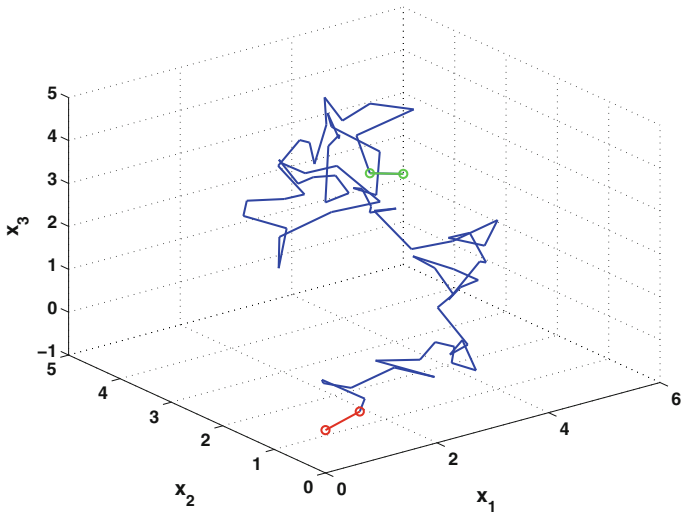

Fig. 1 75 ms trace of a diffusing molecule

If the particles propagate through a three dimensional environment, this movement can be modeled as three independent displacements (one for each dimension) [19] and the total displacement, $\vec{r}$, in one time step can be found as

$$\vec{r} = (\Delta x, \Delta y, \Delta z). \tag{7}$$

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 6, suggests generating Gaussian random numbers with the given parameters for each dimension and at each step independently.

If you just want to simulate a particle movement starting from the origin without replication, you just create the movement at each time step. In each time step, it is normally distributed and can be considered as the accumulation of normally distributed random variables. You can find a 75 ms trace of a molecule depicted in Fig. 1. Trace of a diffusing molecule is produced by the following MATLAB code. The molecule starts diffusing from the origin $(0, 0, 0)$. First and the last steps are marked.

```
function [ trace ] = ...
    diffusion_sim3D_pointSource_singleMolecule(...
    D,...
    sim_time  ...
    )

% D  :Diffusion coefficient in micro meter^2 / seconds
% sim_time  :Duration of the simulation in seconds
```

```
% Time step is 1 ms
delta_t = 0.001;
sim_step_cnt = floor(sim_time / delta_t);

% Standard deviation of step size N(0,sigma)
sigma = (2*D*delta_t)^0.5;

steps = normrnd (0, sigma, sim_step_cnt, 3);

% Trace is just the cumulative sum of steps
trace = cumsum(steps);

end
```

If you want to simulate the diffusion of many particles released from the origin, and want to find the concentration at a distance depending on the time, you can approximate it with the following MATLAB code. For more exact results you should modify the code and choose Δt as small as possible. In this code snippet, it is assumed that counting the molecules passing from the predetermined distance gives the concentration at that distance if you choose Δt small enough.

```
function [ time_line ] = ...
   diffusion_sim3D_pointSource_timeHistogram(...
   r,  ... Distance in micro meters
   D,... Diffusion coefficient in micro meter^2 / seconds
   numMolecules,  ... Number of released molecules
   sim_time,  ... Duration of the simulation in seconds
   replication  ... Replication count
   )
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function simulates simple diffusion process
% and finds the C(r,t) estimate at a distance
% No Receiver/Reception is assumed
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
delta_t = 0.001;
sim_step_cnt = floor(sim_time / delta_t);

% Standard deviation of step size N(0,sigma)
sigma = (2*D*delta_t)^0.5;
mol_position1 = zeros (numMolecules, 3);

time_line = zeros (1, sim_step_cnt);
for i=1:replication
  for t=1:sim_step_cnt
    rem_mol_cnt = size(mol_position1,1);

    % propagate the molecules via diffusion
    mol_displace = normrnd (0, sigma, rem_mol_cnt, 3);
    mol_position2 =  mol_position1 + mol_displace;
```

```
        % evaluate the previous/current distance
        mol_rPrev = sum( mol_position1.^2, 2 ).^0.5;
        mol_rCurr = sum( mol_position2.^2, 2 ).^0.5;

        % find the molecules passing from the r
        r_mask_1 = mol_rPrev < r & mol_rCurr > r;
        r_mask_2 = mol_rPrev > r & mol_rCurr < r;

        % record the time_line
        time_line(t) = time_line(t) + ...
                       nnz(r_mask_1) + nnz(r_mask_2);

        mol_position1 = mol_position2;
      end
    end

    time_line = time_line / replication;

    end
```

For the reception process, we should consider small changes in the simulator code depending on the reception process and the environment. To simulate a CvD system, one needs to define the reception process.

2.1.1 Simulation of CvD

If we consider the CvD system, we should also consider the reception process at the receiver side. In nature reception process is done via ligand-binding and the molecule hitting to the receiver is absorbed and removed from the environment. If we are considering the absorption process at the receiver side, considering the $C(x, t)$ formulation is not the correct way for finding the hitting time histogram.

Therefore we need to consider the absorption process and if we consider the case when there is an absorber, the analytical solution gets complicated for higher dimensions, and adding reflectors in the environment makes it even tougher. One can find the time independent absorption probability in the long run via utilizing the symmetry and the image method. However, in communication theory we need the time distribution of particle absorption we can not have an infinite length symbol duration to reach steady state. While dealing with absorbers, the dynamics of the process changes and is named as the First Passage Process (FPP) [18].

In the 1-D and 2-D environments, diffusing particles hit the receiver in the long run with probability 1 (recurrent process). However, when we consider the 3-D environment, there is a nonzero surviving probability for a diffusing particle [18]. In the 1-D environment, the first hitting probability is

$$f_{hit}(r_0, t) = \frac{r_0}{\sqrt{4\pi D t^3}} e^{-r_0^2/4Dt} \tag{8}$$

where r_0 is the distance to the absorber point. First hitting probability in the 1-D environment is inversely proportional in $t^{3/2}$. In the 1-D environment, we have the closed form solution for first hitting probability function. However for the 2-D or 3-D environments, even with a symmetrical receiver, the closed form solution is a hard surface integration or differential equation problem. Hence, we simulate the diffusion channel and the reception process. If we want to simulate the diffusion with an absorber receiver, we define the reception as removing the hitting molecule from the environment. Hence, we move the molecules in the 3-D environment according to diffusion dynamics, remove and record the molecules when they hit the receiver.

For simulating the basic scenario of a point source transmitter and an adsorbing spherical receiver, we can extend the basic diffusion of multiple particles. Here we give a basic MATLAB code for particles being released from a point source at the origin $(0, 0, 0)$ with a receiver of radius r_{rcv} located at a distance d. Any molecule coinciding with the receiver volume is regarded as a received molecule and removed from the environment.

```
function [ rcv_molecule_histogram ] = ...
   diffusion_sim3D_pointSource_sphereRcv_timeHistogram(...
   d,  ... Source-receiver separation in micro meters
   r_rcv ... Radius of the receiver body
   D,... Diffusion coefficient in micro meter^2 / seconds
   numMolecules,  ... Number of released molecules
   sim_time,  ... Duration of the simulation in seconds
   )
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function simulates a simple CvD mechanism using
% the diffusion process for an adsorbing point source and
% sphere receiver and finds the histogram of the number
% of received molecules
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
delta_t = 0.001;
sim_step_cnt = floor(sim_time / delta_t);

% Standard deviation of step size N(0,sigma)
sigma = (2*D*delta_t)^0.5;

% Molecule release point is at (0, 0, 0)
mol_position1 = zeros(numMolecules, 3);

% Location of the center of the receiver body
center_rcv = [d + r_rcv, 0, 0];

rcv_molecule_histogram = zeros (1, sim_step_cnt);
for t=1:sim_step_cnt
rem_mol_cnt = size(mol_position1, 1);

% propagate the molecules via diffusion
mol_displace = normrnd(0, sigma, rem_mol_cnt, 3);
mol_position2 =  mol_position1 + mol_displace;
```

```
    % calculate the pairwise difference between the location of
    % each molecule and the center of the receiver
    pairwiseDiff = bsxfun(@minus, mol_position2, center_rcv);

    % euclidean distance of molecules to receiver center
    dist_2_rcv = sqrt(sum(pairwiseDiff.^2, 2));

    % find the received molecules
    rcv_mask = dist_2_rcv < r_rcv;

    % record the histogram
    rcv_molecule_histogram(t) = nnz(rcv_mask);

    % remove received molecules from environment
    mol_position1 = mol_position2(~rcv_mask, :, :);
    end

    end
```

Although the code itself is very simple, when simulating a complete communication environment, we may need to perform tens of thousands of molecule releases, each release representing bits or symbols. When this happens, the number of particles roaming free in the communication medium gets very large, and various tricks are needed for a shorter simulation duration.

The first trick is to constrain the size of the communication medium by creating a very big bounding region around the communicting pair. This way, if any molecules reach very far, we can eliminate and remove those molecules from the environment since the probability that they will make it back to the receiver is very low. By the nature of the diffusion process, the longer the communication is carried on, the farther apart the molecules released early on will go. A very large constraining region will have a very small effect on the simulation performance and a very small region constraint will keep molecules from taking the so-called 'scenic route' and reaching the receiver. Therefore, the size of the constraining region should be chosen carefully.

Another improvement can be made by choosing the simulation time step Δt adaptively depending on the closeness to the receiver. The molecules are moved at every time step, hence they have discrete movements and we just consider the final locations to decide whether it is absorbed or not. Therefore, some of the molecule movements, those actually hit to the receiver, can be falsely categorized due to last location being outside the receiver. To resolve this issue, whether implementing the test for line segment (path) and receiver sphere intersection should be considered or choosing Δt as small as possible is encouraged. Choosing Δt as a small increment may increase the run time drastically, hence we also encourage to use region adaptive Δt values. If a molecule is close to the receiver its time step should be small, on the contrary if a molecule is far away from the receiver its time step may be chosen a larger value and it waits for other molecules.

Another notable improvement can be made is by placing the receiver's center at the origin and releasing the molecules from the point $(-(d+r_{rcv}), 0, 0)$. This way, the

line of code where we calculate the pairwise difference between the location of each molecule and the center of the receiver will become obsolete. Moreover, instead of calculating the Euclidean distance of the molecules and the receiver's center, we may use the Euclidean squared distance for comparison. This way, we will not need to perform a square root operation in every simulation step, but will detect the received molecules by comparing the sum of squares of the molecule locations with r_{rcv}^2.

This basic point source transmitter—spherical receiver simulation can also be easily converted to one having a spherical transmitter. In this case, it is crucial to implement the characteristics of the transmitter. If the transmitter is also adsorbing, then the reception mechanism should also be implemented at the transmitter side. If the transmitter is not adsorbing, then the molecules trying to diffuse towards the inside of the transmitter should be blocked by the transmitting body. There are several ways of implementing transmitter blockage. The first and easiest way is to roll back the movement of any molecules that end up inside the transmitter body at the end of a simulation step. These molecules can be thought of as staying still for a single simulation step. Another way is to re-draw the molecule displacement from the normal distribution. However, this choice of blockage simulation will result in an immense number of trials at the start of molecule release since the molecules are most likely to end up inside the transmitter early on in the simulation. One other choice is flipping the sign of the molecule displacement vector to move the molecule to the opposite direction, therefore preventing it from stepping into the transmitter body.

2.1.2 Deciding Replications

One of the most crucial factors in running simulations is the number of replications (runs) with different random seeds in order to rule out the effect of random number generation. If we consider the actual implementation of a system as a population and the simulation runs as the samples from that population, according to the Central Limit Theorem, we must run the simulations at least with 30 different random seeds for each of the different random aspects of the simulation. However, if the sample sample size is less than 30, the Central Limit Theorem will work if the distribution of the population is not severely nonnormal [8]. In this section, we compare the results of 30 different sample sizes against smaller ones to see if there is a statistically significant difference inbetween. If there is no difference we can conclude that we do not have to take 30 different runs, so we can get the same results with smaller samples. The main random behaviours in the simulation of CvD are movements of the molecules and the bit sequence that the transmitter transmits to the receiver.

To compare the significance of different sample sizes, we use student t-test. Our null hypothesis is that the mean data rate of n samples ($n < 30$) is equal to mean data rate of 30 samples. For this purpose, we compare the mean data rates of all samples from 2 to 28 with 30 samples. Although we could not reject the null hypothesis in any of these tests, the p values of these tests differ from each other. The technical definition of the p-value is the smallest level of significance that would lead to rejec-

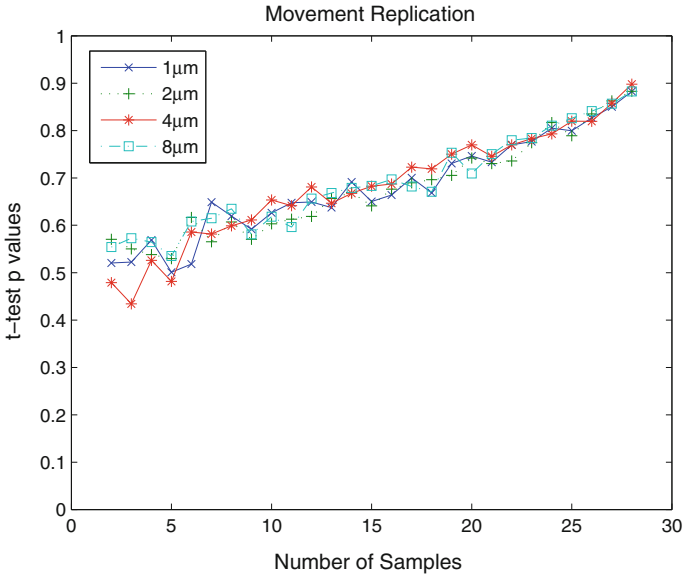

Fig. 2 Comparison of p-values for movement replication

tion of the null hypothesis [8]. Therefore, we can say the p-value is the power of the test in the sense that, as the p-value increases the test becomes more reliable.

Movement Replication

In Fig. 2, we present the comparison of the p-values for the replication of the movement of messenger molecules. We can conclude from the figure that the effect of the number of samples to the mean of the data rate is independent from the distance between the transmitter and the receiver. Moreover, as the number of samples increases (approaching 30), the p-value approaches to 1 as expected. As stated earlier, even for two samples the test does not reject the null hypothesis, i.e., the mean of the small sample is equal to the mean of the 30 samples. However, since the result is not as strong as larger number of samples, the researcher must decide how many samples would be sufficient regarding the power of the test with different sample sizes.

Bit Sequence Replication

In Fig. 3, similar to the movement replication, we observe that the effect of the number of samples is independent from the distance. Again, for bit sequence replication even two samples are sufficient for not-rejection of the null hypothesis, but the power of the test increases as the number of samples increases. The researcher must decide that how many sample are needed regarding the power of the test.

In conclusion, two different sample sizes must be selected for the implementation, one for replication of molecule movements and one for replication of bit sequences.

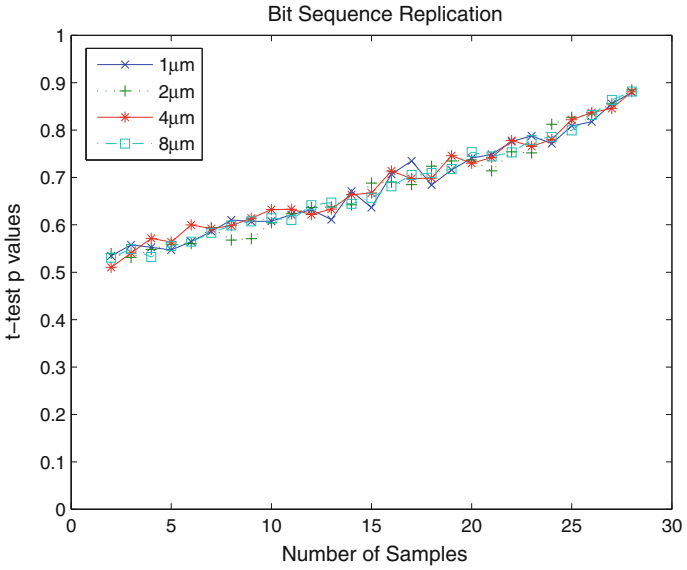

Fig. 3 Comparison of p-values for movement replication

Note that, if n is the selected number of samples for molecule movement replications and m is the number of bit sequence replications, in total $n \times m$ runs have to be run for all combinations of movement and bit sequence random seeds.

3 Simulation of Complex CvD Systems

3.1 Transmission and Reception Enhancements

Due to the Brownian motion characteristics in the diffusion environment, not all molecules reach the receiver body. Many of them scatter away, especially when the transmitter and receiver bodies are farther apart. As shown in [14], the distance between the transmitter and receiver bodies is a key factor that affects the propagation of molecules and increasing this separation degrades the reception performance substantially. Thus, enhancements at both the transmission and reception stages of 3-D diffusion are crucial for sustaining a successful communication. To this end, we present two manners of enhancements that can be implemented at either the transmission and/or the reception stage.

3.1.1 Reception Enhancement with Protrusions

The first enhancement over CvD can be achieved by equipping the receiver with protrusive elements extending towards the transmitter. In this manner, the number of messenger molecules reaching the receiver in a given amount of time can be increased, providing higher quality communication.

Biological Background

In biology, protrusions are described as arm-like cytoplasmic projections extending from the cell membrane. Many cells in the nature are equipped with the protrusion mechanism for enhancing their operations. The enhancements to the basic cell operations provided by protrusions include expanding the surface area, improving chemical communication with neighboring cells, and cell motility. By using protrusions, cells can enable several cellular processes like wound healing, embryonic development, and neuronal growth-cone pathfinding [16]. There is a multitude of protrusive elements in the nature such as microvilli on the surfaces of epithelial cells, stereocilia (hair cells enabling hearing) in the inner ear, or lamellopodia and filopodia found in fibroblasts (connective tissue cells) [5]. The type of protrusions that we mention here for enhancing CvD are based on the filopodia type protrusions.

The working principle of the protrusions is a continuous and ongoing process in the basic operations of a cell. A cell continuously monitors the environment it is in, always looking for signals that can arrive from the environment. These signals can be physical, chemical, or in some cases even luminous. A signal from the environment is detected by the receptors located on the cell membrane. Upon detection of a signal, another cellular process is triggered and the existence of a signal is transmitted to the cell interior. A cell equipped with the protrusion ability begins extending its protrusions once the internal signaling begins. Depending on the type of the cell and the type of the received signal from the environment, protrusions can be extended on the whole surface of the cell, or can also be directed towards the source of the external signal.

The filopodal protrusion mechanism in cells is driven by a branched network of actin filaments found in the cytoplasm. Once the decision of extending protrusions is made inside the cell, motor proteins called Myosin-X bundle polymerized actin filaments together and start pushing the cell membrane outwards [16]. To hold actin filaments together, Fascin protein is utilized and Dia2 proteins monitor the actin polymerization at the elongating tip. The base of the protrusion structure holds a cross-linked form for support, so that the actin filaments will not sink back into the cytoplasm. Steps of the protrusion extension mechanism can be observed in Fig. 4. We also note that while the protrusion is being extended, the external signal triggering this process is still continuously monitored with the receptors on the protruding cell membrane. In the nature, the filopodal protrusions may extend up to 40 μm in length and have a radius of several hundred nanometers. The versatility of protrusion size makes it a very useful tool in enhancing cellular operations.

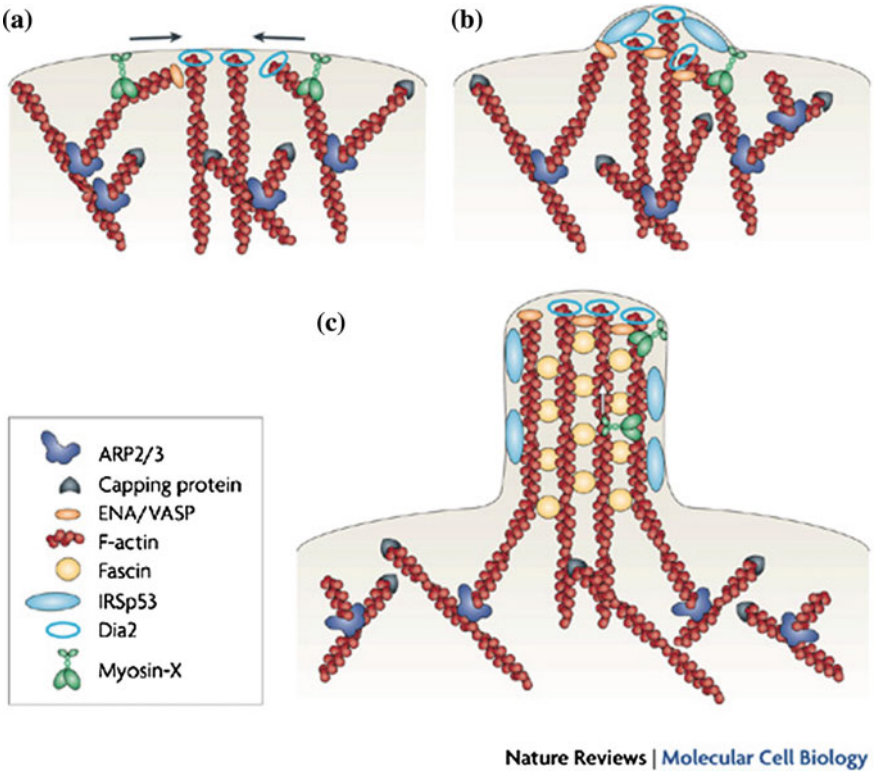

Fig. 4 The working model for filopodia formation [16]. **a** Myosin-X proteins pull actin filaments together towards the protrusion site. **b** Actin filaments push against the cell membrane. **c** Fascin proteins link the actin bundles to form a strong protrusion

CvD with Protrusion Enhancement

The protrusion mechanism in cells can be adapted to the CvD scenarios such that the receiver body is equipped with protrusions that are capable of messenger molecule reception. There are a multitude of ways for implementing protrusions on the receiver surface. The two most important points in protrusion enhancement are the deployment strategy on the surface and the geometric shape of these extensions. Protrusions can be deployed on the whole surface of the receiver either randomly or in a uniform manner. The deployment can also be made such that the part of the receiver facing the transmitter source is favored. Protrusions can be extended fully perpendicular to the receiver membrane, or they can face the transmitter source directly. Moreover, the geometric shape of the protrusions can be either conical or cylindrical. Each combination of deployment, direction, and geometric shape has various advantages and drawbacks. For example, protrusions favored on the transmitter-facing side of a receiver are very useful for capturing messenger molecules earlier, but the passerby molecules are ignored on the other sides of the receiver. A uniform distribution on the

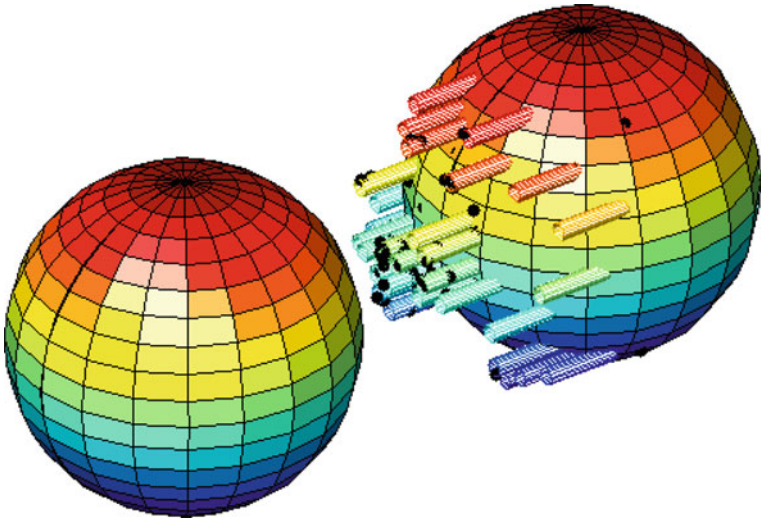

Fig. 5 Example communication system setup with protrusions and received molecules [11]

whole surface has reception from all sides, but does so at the expense of messenger molecule reception time.

As an example, we investigate the scenario where the protrusions are shaped into cylinders extending from a sphere-shaped receiver surface and directly facing a sphere-shaped transmitter. These cylinders have radii r_{prot} and length h_{prot}. It is very important to conserve the total volume of the receiver when extending protrusions for making sound comparisons between classical CvD and protrusion enhanced CvD approaches. Therefore, we conserve the total volume of the receiver NeN by extracting the volume occupied by the protrusions from the backside of the receiver.

Figure 5 shows an example communication system with a few tens of protrusions and some messenger molecules received either by the receiver surface or the protrusions. The protrusion sites are chosen at random, with the constraint that they face the transmitter NeN. The messenger molecules are removed from the environment once they come in contact with either the receiver surface or the protrusions. This way, the reception of messenger molecules is improved by decreasing the effective distance between the transmitter-receiver pair.

Performance Evaluation

We investigate the effect of protrusions on the CvD performance by simulations. In this setup, we have a spherical transmitter-receiver pair, where the receiver is equipped with protrusions facing the transmitter. The release of messenger molecules on the transmitter body is made from a single point located on the transmitter surface, directly facing the receiver. The performance metric we use is the probability of a messenger molecule hitting the receiver in a given amount of time t_s, called the symbol duration. The aim is to receive as many molecules as possible in a single

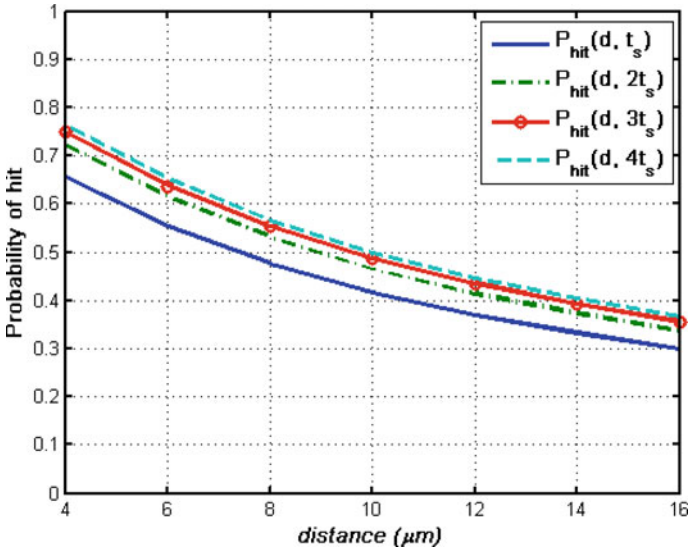

Fig. 6 Probability of hitting the receiver versus various transmitter-receiver pair distances without protrusion [11]

symbol duration, so that upcoming messages on consecutive symbol durations do not hamper each other's operations.

In Figs. 6 and 7, we observe the effect of protrusion on hitting probabilities for various amounts of distances between the transmitter and the receiver. The x-axis denotes the distance between the transmitter and receiver bodies, and the y-axis denotes the probability of a messenger molecule being received by either protrusions or the receiver surface. Each curve represents the probability of hit in a duration of 1–4 t_s. We observe that using protrusion enhances the probability of receiving a messenger molecule significantly. Moreover, the gaps between the curves are smaller, which means that the messenger molecules lagging behind and creating inter-symbol interference are reduced. Decreasing the inter-symbol interference and increasing the hitting probabilities play a crucial role in successful communication. We also observe that, keeping the protrusion length at 3.5 μm, the advantage obtained from extending protrusions decreases, and the gap symbolizing the inter-symbol interference increases.

3.1.2 Tunnel-Based Approach Using Destroyer Molecules

A second type of enchancement over CvD communication system can be achieved by shaping the molecular signal. This idea focuses on manipulating the molecular communication channel between the transmitter—receiver couple using a secondary type

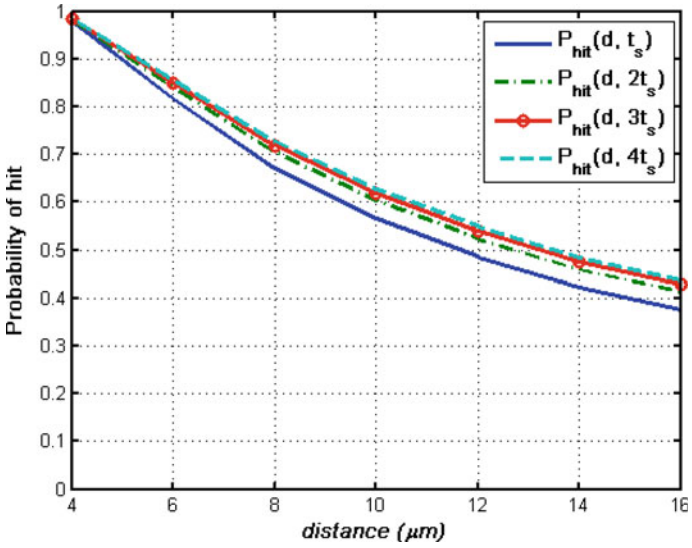

Fig. 7 Probability of hitting the receiver versus various transmitter-receiver pair distances with protrusion, $r_{prot} = 0.5\,\mu\text{m}, h_{prot} = 3.5\,\mu\text{m}$ [11]

of molecules, called Destroyer Molecules (DM), in an effort to shape the received signal according to the desired signal structure.

Biological Background

Neuromuscular junction (NMJ) is one of the occurrences in biological systems where two cells communicate with each other using an intermediary molecule that propagates in the extracellular environment following diffusion dynamics.

NMJ resides between a nerve and a muscle cell couple. When the muscle needs to be contracted, the nerve cell releases pre-synthesized special neurotransmitter molecules, called Acetylcholine (ACh), into the NMJ. These molecules propagate in this environment and when they get close to the cell membrane of the muscle cell, they bond with the transmembrane receptors, called the ACh receptors (AChR). The neurotransmitters stay in the bounded state for some time after which the bond degrades and the ACh molecules are again set free to the NMJ.

As seen in Fig. 8, the NMJ is a semi-closed environment and the molecules inside usually move between the two cells. Hence, after the degradation of the bonds between ACh and AChR, the neurotransmitter molecules are highly likely to re-bond with the receptors. Such re-bondings cause further unwanted muscle contractions, and after a few muscle contraction signals, the NMJ can be filled with ACh molecules. Thus all further contraction signals will be blocked after several contractions. To keep the communication between the nerve and muscle cell couple possible, the ACh molecules in the environment should be removed from the NMJ after the muscle cell is successfully contracted. This cleaning process is achieved through the use of a

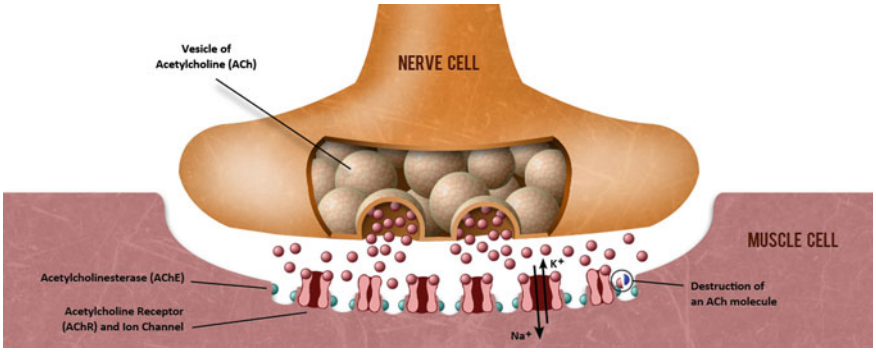

Fig. 8 Neuromuscular junction [13]

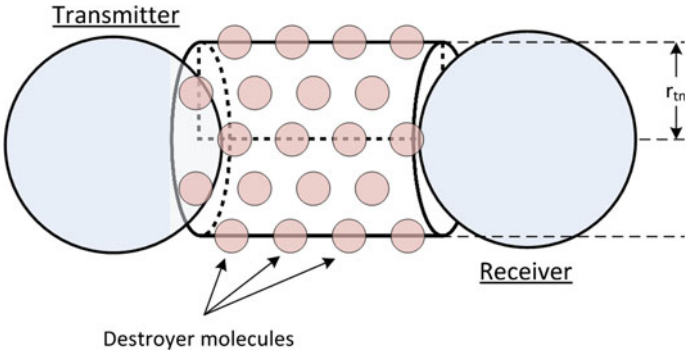

Fig. 9 Cylindrical tunnel environment for CvD [13]

secondary type of molecule, called Acetylcholinesterase (AChE). The AChE molecules interact with the ACh molecules and remove them from the NMJ by hydrolizing ACh into its basic components, acetate and choline. Thus, AChE molecules enable the muscle cell to be capable of responding to further contraction signals.

Tunnel Structure

If we approach the NMJ system from a communication perspective, the receiver can be considered utilizing specialized so-called Destroyer Molecules (DM) to control the shape of the signal and eliminate the undesired components of a signal. Also, the usage of these molecules reduces or eliminates the effects of the Intersymbol Interference (ISI) and allows the selection of shorter symbol durations. From a topological point of view, the destroyer molecules can be deployed in the environment in many different ways. As an example, we investigate the case where the DMs are deployed to form a cylindrical tunnel-like structure that forms a spatially restricted path between the transmitter and the receiver (Fig. 9). This structure follows biological extensions like filopodia and cytoneme. We assume that, when a messenger

molecule hits a DM on this tunnel, it is assumed to be destroyed and removed from the environment.

The main idea behind this type of deployment is to get rid of the stray messenger molecules in the environment so that only the ones that contribute to the spike in the reception time distribution remain while other molecules are eliminated in the environment. As in the case of AchE, we assume that DMs are bigger in size compared to the messenger molecules and they are connected to one of the communicating pair through other DMs. We also assume that they are immobile in the environment. Due to the chemical attraction between the messenger and the DMs, when a messenger molecule gets close to a DM, it is attracted by the destroyer and removed from the environment. The cylindrical tunnel has a radius of r_{tn}, which is a variable in the simulations. In order to simplify the analysis, we assume that the whole tunnel is composed of DMs and any molecule that diverts from the shortest path between the transmitter and the receiver more than r_{tn} is destroyed.

Performance Evaluation

We simulate a Single Transmitter-Single Receiver topology using Monte Carlo simulations, whose parameters are as given in [13]. We evaluate the communication capability of this deployment scenario using two performance metrics: the hitting time at the receiver (T_{hit}) and the corresponding data rate of the CvD system based on r_{tn} and d, the distance between the transmitter and the receiver.

As seen in Figs. 10 and 11, according to the selected metrics, we can clearly see that a tunnel-like structure increases the communication capability of a CvD system. The average T_{hit} value decreases by selecting a tight tunnel radius (r_{tn}). Specifically, choosing r_{tn} close to d reduces the average T_{hit} value roughly ten times. This is due to the elimination of slow moving molecules from the environment by the DMs.

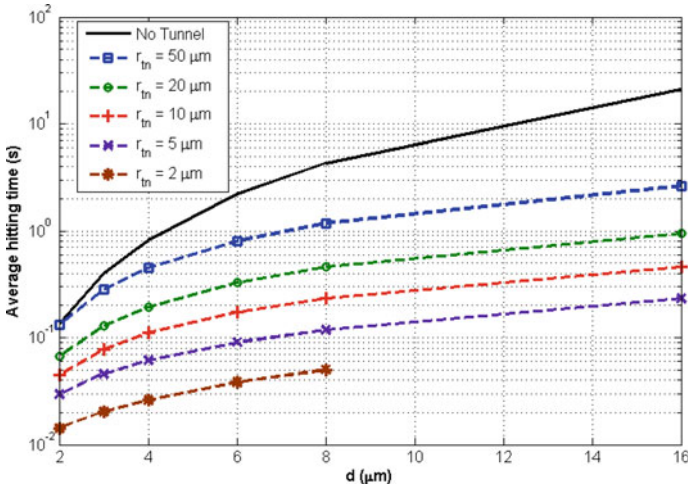

Fig. 10 Effect of destroyer molecules on average hitting time [13]

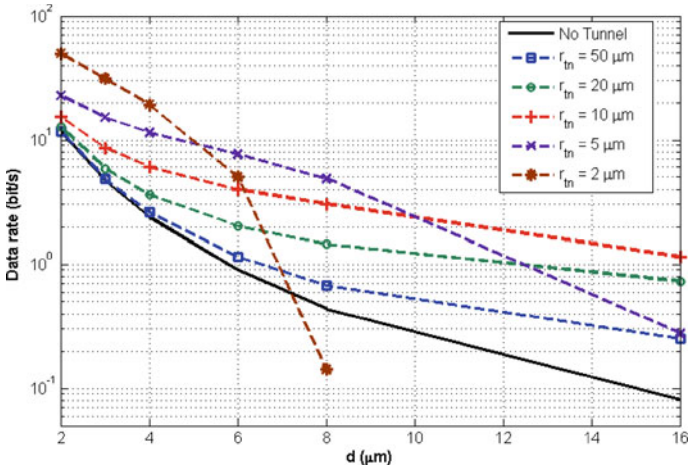

Fig. 11 Effect of cylindrical tunnel environment on data rate [13]

The last curve ($r_{tn} = 2$ m) does not yield any results since no molecules arrive at the receiver when the d/r_{tn} value is too high (i.e., all of them are destroyed due to the narrowness of the tunnel).

Furthermore, the data rate of the system considerably increases with the usage of the tunnel. This is also more prevalent with increasing d values. The d/r_{tn} limit is also apperant in this metric. Beyond $d = 4m$, the data rate of the tunnel environment with $r_{tn} = 2$ m drops very quickly. Similar to the average T_{hit}, this drop is due to the fact that in such environments the tunnel becomes too tight and the molecules hit the receiver with a very small probability.

Based on these results, it is apperant that this tunnel-like deployment of DMs greatly increases the performance of the CvD channel. However, in a realistic environment, the question of how a tunnel structure can be constructed between the communicating pair arises. Such a tunnel will also be costly in terms of energy when considering the energy cost of synthesizing the DMs. More relaxed and less restrictive DM deployment schemes must be investigated.

3.2 Distributed Simulations

Due to the need for simulating large number of nano-scale objects, the scalability of the simulation of molecular communication is an important concern. Flexibility, interoperability, and reusability are other design criteria for the simulation of molecular communication. Selecting the right architecture, which supports software component reuse and high level of scalability enables a growing library of simulation components. Distributed simulation can enable simulation of complex scenarios. A distributed simulation is a collaborative system in which each simulation unit runs

on an independent computational unit and communicate to simulate a scenario in a commonly managed logical time. High Level Architecture (HLA) [1], which is the IEEE standard for distributed simulations, is a promising option for implementation of distributed molecular communication for large scale simulators.

3.2.1 High Level Architecture

HLA was developed by the United States Department of Defense (DoD) to cover defense applications. After its increasing use by other industries and research areas, it was standardized by the IEEE [1]. HLA defines the component model and their interactions. The component model contains federates, which communicate over Run Time Infrastructure (RTI). Federates enable abstraction, and they form a simulation model referred to as federation. This abstract component model enables independent design and development of components and also distributed execution of the simulation.

The RTI is the backbone of the federation, and provides synchronization, communication, and data exchange services to the federates [3]. Each federate can be an independent event or time driven simulations, real time simulation with human interaction, live system or equipment. 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 [7]:

1. *Federation management:* Basic functionality required to create and execute a federation.
2. *Declaration management*: Management of data exchange between federates, using the information provided by federates.
3. *Object management*: Creation, deletion, identification, and other services at the object level.
4. *Ownership management:* The dynamic transfer of ownership of object/attributes during an execution.
5. *Time management:* Synchronization of runtime simulation data exchange.
6. *Data distribution management*: Routing of data among federates during federation execution.

The implementation of these services are not in the scope of HLA interface specification. The specification only focuses on the way these services are accessed.

3.2.2 A Distributed Simulation Design

Using HLA, it is possible to create a simulation design that focuses on interoperability, re-usability, and scalability. With such a design, 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 sim-

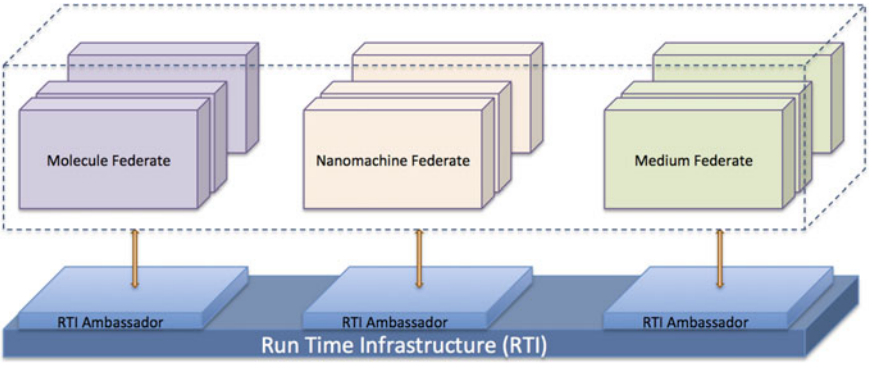

Fig. 12 A distributed simulation architecture

ulations on distributed computer systems. There are four primary principal benefits of executing a simulation program across multiple computers [10]:

1. *Reduced execution time.* It is possible to decrease execution time by dividing simulation computation into smaller sub-computations, and distributing these computational tasks to different computational units,
2. *Geographical distribution.* Running the simulation on geographically distributed computers enables interaction with users or live systems which have to be in different locations.
3. *Integrating simulators that execute on machines from different manufacturers.* A distributed approach enables interaction between different platforms. This approach enables the use of hetoregenous hardware and software system to execute a simulation.
4. *Fault tolerance*. If a unit fails while using multiple computational units, another unit can pick up the load of the failing unit.

A possible distributed architecture [4] is shown in Fig. 12. The simulation is defined as a Molecular Communication Federation. Separate federates are defined for molecules, nanomachines and the medium. The federates communicate with each other over RTI. 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.

The Molecule Federate models the molecules in the environment and is responsible for the movements of the molecules. The molecules propagate in the three dimensional space. This movement can be modeled as three independent movements in each dimension as described in Sect. 2.1. The environmental parameters are defined in medium slices. The Molecule Federate subscribes to the Medium Federate attributes. Hence, when a molecule moves into another medium, the parameters are communicated to the Molecule Federate by RTI.

The Nanomachine Federate abstracts the nanomachines in the environment and is responsible for the transmission and the reception mechanisms. Different nanoma-

chine federate implementations can define different transmitter and receiver behaviours. Alternative transmitter nanomachine implementations can release molecules instantaneously or sequentially, from a single point on the surface, or from multiple points. Similarly, different implementation of a receiver nanomachine can receive molecules all over its surface, or an alternative implementation can receive only via receptors distributed on its surface.

The Medium Federate abstracts the medium slices that represent subsets of the 3-dimensional space for simulation. The collision handling for molecules can be implemended in the Medium Federate. Based on the model to be simulated, different collision management implementations can be implemented, which may consider underlying physical and chemical laws during a collision process. Simulation scalability can be achieved by assigning different medium slices to different medium federates.

3.2.3 Performance Evaluation

For the performance evaluation of the distributed simulations, the speedup (S) can be defined as

$$S = \frac{T_s}{T_m} \tag{9}$$

where T_s is the execution time with a single node and T_m is the execution time with multiple nodes. Linear speedup is achieved when speedup is equal to the number of nodes used in the execution. Different algortihms or architectures for distributed execution of simulations for molecular communication can be compared using speedup metric, and optimal architecture can be selected.

4 Conclusions

This chapter approaches diffusion from a communication engineer's perspective and provides the researcher some useful hints and intricacies for designing molecular communication simulations in an effective manner. These hints and intricacies include the selection of the number of replications required in terms of movement and bit sequence for fast but meaningful results. We utilize this information further by discussing the design of more complex CvD systems such as tunnel-based approach that utilizes destroyer molecules and distributed simulator design based on HLA.

Acknowledgements This work has been partially supported by the 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 the Scientific and Technical Research Council of Turkey (TUBITAK) under Grant Number 112E011. M. Şükrü Kuran partially carried out the work presented in this paper at LINCS (http://www.lincs.fr).

References

1. Modeling and simulation (m & s) high level architecture (hla) (IEEE 1516-2010 series). http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5553438. Accessed 29 Jan 2014
2. N3sim is a simulation framework for diffusion-based molecular communication in nanonetworks. http://www.n3cat.upc.edu/n3sim. Accessed 29 Jan 2014
3. AbouRizk S, Mohamed Y, Taghaddos H, Saba F, Hague S (2010) Developing complex distributed simulation for industrial plant. In: Proceedings of the 2010 winter simulation conference, pp 3177–3188
4. Akkaya A, Genc G, Tugcu T (2014) HLA based architecture for molecular communication simulation. Simul Modell Pract Theory. doi:10.1016/j.simpat.2013.12.012, http://www.sciencedirect.com/science/article/pii/S1569190X13001925
5. Ananthakrishnan R, Ehrlicher A (2007) The forces behind cell movement. Int J Biol Sci 3(5):303
6. Berg HC (1993) Random walks in biology. Princeton University Press
7. Dahmann JS (1997) High level architecture for simulation. In: First International workshop on distributed interactive simulation and real time applications, 1997, p 32
8. Douglas C, Montgomery GCR (2006) Applied statistics and probability for engineers. Wiley
9. Felicetti L, Femminella M, Reali G (2012) A simulation tool for nanoscale biological networks. Nano Commun Netw 3(1):2–18
10. Fujimoto RM (2000) Parallel and distributed simulation systems. Wiley
11. Genc G, Yilmaz HB, Tugcu T (2013) Reception enhancement with protrusions in communication via diffusion. In: 2013 First International Black Sea conference on communications and networking (BlackSeaCom), pp 89–93. IEEE
12. Gul E, Atakan B, Akan OB (2010) Nanons: a nanoscale network simulator framework for molecular communications. Nano Commun Netw 1(2):138–156
13. Kuran MŞ, Yilmaz HB, Tugcu T (2013) A tunnel-based approach for signal shaping in molecular communication. In: 2013 IEEE international conference on communications workshops (ICC), pp 776–781
14. Kuran MŞ, Yilmaz HB, Tugcu T, Özerman B (2010) Energy model for communication via diffusion in nanonetworks. Nano Commun Netw 1(2):86–95
15. Llatser I, Pascual I, Garralda N, Cabellos-Aparicio A, Alarcon E (2011) N3sim: a simulation framework for difusion-based molecular communication. In: IEEE Technical Committee on Simulation, vol 8
16. Mattila PK, Lappalainen P (2008) Filopodia: molecular architecture and cellular functions. Nat Rev Mol Cell Biol 9(6):446–454
17. Moore M, Enomoto A, Nakano T, Suda T, Kayasuga A, Kojima H, Sakakibara H, Oiwa K (2006) Simulation of a molecular motor based communication network. In: Bio-inspired models of network, information and computing systems
18. Redner S (2001) A guide to first-passage processes. Cambridge University Press
19. Saxton MJ (2007) Modeling 2d and 3d diffusion. In: Methods in membrane lipids. Springer, pp 295–321
20. Toth A, Banky D, Grolmusz V (2011) 3-d brownian motion simulator for high-sensitivity nanobiotechnological applications. IEEE Trans Nanobiosci 10(4):248–249
21. Tyrrell HJV, Harris K (1984) Diffusion in liquids. A theoretical and experimental study. Butterworth Publishers, Stoneham, MA