

N3Sim: Simulation Framework for Diffusion-based Molecular Communication Nanonetworks

Ignacio Llatser^a, Deniz Demiray^{a,b}, Albert Cabellos-Aparicio^a,
D. Turgay Altılar^c, Eduard Alarcón^a

^a*Nanonetworking Center in Catalunya (N3Cat), Universitat Politècnica de Catalunya
Jordi Girona 1-3, Campus Nord D6-008, 08034 Barcelona, Spain*

Email: {llatser,acabello}@ac.upc.edu, eduard.alarcon@upc.edu

^b*Istanbul Technical University Informatics Institute*

ITU Ayazaga Campus, 34469, Maslak, Istanbul, Turkey

Email: demirayde@itu.edu.tr

^c*Istanbul Technical University Faculty of Computer & Informatics*

ITU Ayazaga Campus, 34469 Maslak, Istanbul, Turkey

Email: altilar@itu.edu.tr

Abstract

Diffusion-based molecular communication is a promising bio-inspired paradigm to implement nanonetworks, i.e., the interconnection of nanomachines. The peculiarities of the physical channel in diffusion-based molecular communication require the development of novel models, architectures and protocols for this new scenario, which need to be validated by simulation. *N3Sim* is a simulation framework for nanonetworks with transmitter, receiver, and harvester nodes using Diffusion-based Molecular Communication (DMC). In DMC, transmitters encode the information by releasing molecules into the medium, thus varying their local concentration. *N3Sim* models the movement of these molecules according to Brownian dynamics, and it also takes into account their inertia and the interactions among them. Harvesters collect molecules from the environment to reuse them for later transmissions. Receivers decode the information by sensing the particle concentration in their neighborhood. The benefits of *N3Sim* are multiple: the validation of channel models for DMC and the evaluation of novel modulation schemes are just a few examples.

Keywords: Diffusion-based molecular communication, Diffusion, Brownian motion, Nanonetwork, Molecular harvesting

1. Introduction

Nanotechnology, first envisioned by the Nobel laureate physicist Richard Feynman in his famous speech entitled “There’s plenty of room at the bottom” in 1959, is giving rise to devices in a scale ranging from one to a few hundred nanometers. During the last few decades, emerging research areas such as nanoelectronics, nanomechanics and nanophotonics are allowing the development of novel nanomaterials, nanocrystals, nanotubes and nanomachines that promise to revolutionize many fields of science and engineering.

Molecular nanotechnology, popularized by Eric Drexler [1], is a particularly advanced form of nanotechnology based on the ability to build molecular machines by means of mechanosynthesis. Since biology clearly demonstrates that molecular machines are possible (living cells themselves are an example), the manufacture of bio-inspired molecular machines using biomimetic techniques is envisaged in the near future.

Nanotechnology is a multidisciplinary field with almost uncountable potential applications. A few examples are presented next. First, in the biomedical domain, nanoparticles such as dendrimers, carbon fullerenes (buckyballs) and nanoshells are currently used to target specific tissues and organs [2]. Another area where nanotechnology plays an important role is environmental science, where molecular and genomic tools are used to uncover the complexity of the induced defense signaling networks of plants [3]. Finally, in the industrial field, molecular-scale filters and pores with well-defined sizes, shapes, and surface properties allow to engineer better functionality in molecular sieving [4].

The envisaged *nanomachines* are the most basic functional units able to perform very simple tasks at the nanoscale, including computing, data storage, sensing, actuation and communication. Cells are a clear example of living nanomachines. Therefore, even though a complete nanomachine has not been manufactured to date, artificial simple nanomachine are expected to become a reality in the near future.

Because of their tiny size, the operation range of nanomachines is limited to their close nano-environment. In consequence, a huge number of them will be required in order to perform meaningful tasks in a real scenario. These nanomachines will also need to control and coordinate their functions, leading to several research challenges in communication at the nanoscale.

Nanonetworks, the interconnection of nanomachines, have emerged as a novel research field which has attracted the interest of researchers from the

domains of information and communication technology, nanotechnology and biology. Nanonetworks will provide means for cooperation and information sharing among nanomachines, allowing them to fulfill more complex tasks. As a consequence, nanonetworks are envisaged to greatly expand the range of applications of nanotechnology. Sample applications of nanonetworks include Wireless NanoSensor Networks [5] and Graphene-enabled Wireless Networks-on-Chip [6].

Current communication techniques, such as electromagnetic, mechanical or acoustic communication, cannot be directly applied at the nanoscale [7]. Instead, several alternative methods have been proposed to interconnect nanomachines, leading to two novel paradigms to implement communications at the nanoscale: molecular communication [7] and graphene-based electromagnetic communication [8].

In molecular communication, different methods are applied depending on the distance between emitters and receivers. These methods can be classified in three alternatives: short-range (nm to μm), medium-range (μm to mm) and long-range (mm to m). For the *short range*, two methods have been proposed [7]. The first one is molecular signaling, consisting in encoding the information into molecules which are emitted into the medium. The second is based on molecular motors, i.e., protein complexes that are able to transport molecules through microtubules. Two mechanisms have as well been proposed for *medium-range* molecular communication: flagellated bacteria [9] and catalytic nanomotors [10]. Both methods are based on encoding the information in DNA sequences (a DNA packet), which are carried it from transmitter to receiver by using bacteria or nanomotors, respectively. Finally, several techniques have been proposed for the *long range*, such as pheromones, pollen and spores [11].

Among the previous techniques, one of the most widely studied is molecular signaling [12, 13, 14]. In molecular signaling, transmitters suspended in a fluid medium emit molecules according to a release pattern which encodes the transmitted information. The emitted molecules move following an erratic trajectory, due to the collisions between them and the molecules of the fluid. As a consequence of this movement and of interactions among the emitted molecules (such as collisions and electrostatic forces), the emitted molecules diffuse throughout the medium. This diffusion causes the concentration of molecules to propagate and spread throughout the space. Finally, receivers measure the local concentration of molecules at their neighborhood and decode the transmitted information.

The remainder of this paper is organized as follows. In Section 2, we briefly describe diffusion-based molecular communication. Next, Section 3 describes the related work. In Section 4, we outline the architecture of the simulation framework *N3Sim* and Section 5 describes the collision detection implemented in *N3Sim*. In Section 6, we present some sample results in order to illustrate its functionalities. Finally, Section 7 concludes the paper and points to future work directions in this area.

2. Diffusion-based Molecular Communication

Diffusion is the process by which particles suspended in a fluid experience a random endless movement. The basic diffusion process is based on Brownian motion, which is due to interactions among the fluid particles and the suspended particles. The microscopic dynamics of this process are extremely complex. It was Einstein in 1905 who first suggested equations to describe this process from a macroscopic point of view. Einstein showed that the concentration of Brownian particles $c(x, t)$ satisfies the diffusion equation:

$$\frac{\partial c(x, t)}{\partial t} = D \frac{\partial^2 c(x, t)}{\partial x^2} \quad (1)$$

where $D = K_b T / 6\pi R \nu$ is the diffusion coefficient, which depends on the Boltzmann constant K_b , the temperature T , the particle radius R and the fluid viscosity ν . Assuming that Q particles start from the origin at the initial time $t = 0$, the 3-dimensional diffusion equation has the following solution:

$$c(x, t) = \frac{Q}{(4\pi Dt)^{3/2}} e^{-x^2/4Dt} \quad (2)$$

The first moment of the particle movement is zero, meaning that the Brownian particle is equally likely to move to the left as it is to move to the right. The second moment, which expresses the mean squared displacement of the Brownian particles, is given by

$$\overline{x^2} = 2Dt \quad (3)$$

The same equation can be applied to obtain the displacement in the y and z axes, since each dimension has an independent contribution to the displacement. By means of these equations, the stochastic movement of a particle affected by Brownian motion can be accurately modeled.

Diffusion-based Molecular Communication (DMC) aims to model the previously-described scenario of molecular signaling. In DMC, the evolution of the molecular concentration throughout space and time is modeled as a diffusion process. Depending on the scenario, different diffusion models may be used, which can be grouped into two categories: normal diffusion and anomalous diffusion [15].

On the one hand, *normal diffusion* refers to the case when the movement of the emitted molecules can be modeled by Brownian motion [16], which is valid when viscous forces dominate the motion of the emitted molecules and the interactions among them can be neglected. In this case, the movement of each of the molecules is uncorrelated and the diffusion process can therefore be macroscopically modeled with Fick’s laws of diffusion [17].

On the other hand, *anomalous diffusion* appears when interactions among the emitted molecules affect their diffusion process. For instance, in a particular case of molecular signaling known as calcium signaling [18], based on the use of positive-charged calcium ions (Ca^{2+}), the electrostatic forces among these ions impact their diffusion process. Another example of anomalous diffusion includes the case when the concentration of emitted particles is very high and the collisions among them affect their movement, a scenario known as *collective diffusion* [19]. These effects may increase or reduce the displacement of suspended particles, processes known as superdiffusion and subdiffusion, respectively. In these cases, the diffusion process can no longer be modeled by Fick’s laws of diffusion and it needs to be analyzed with other methods, such as correlated random walk [20].

3. Related Work

Several authors have developed analytical models of the physical channel of DMC [21, 22, 23, 24]. A recent study considers molecules as discrete resources, which can be reused for multiple transmissions by means of molecular harvesting [25]. However, to the best of our knowledge, a validation of these models is missing. In order to validate these theoretical models, either an experimental study or simulations are needed. Despite recent advances in synthetic biology, an experimental setup of molecular communication is still very challenging to build; consequently, simulation currently seems the most feasible choice. A simulator would allow to recreate an environment of DMC and measure the relevant metrics to evaluate its performance, such as channel attenuation, delay and throughput. The simulator outputs could be

then compared with the results from the analytical models in order to assess their validity and to create novel, more accurate models.

The field of molecular dynamics considers the simulation of the physical movements of atoms and molecules in a given scenario. Molecular dynamics simulators typically consider very accurate models of atoms and molecules and they are highly scalable to run in supercomputers with thousands of computing cores. Some popular simulators include NAMD [26], a parallel molecular dynamics code designed for high-performance simulation of large biomolecular systems; GROMACS [27], able to simulate the Newtonian equations of motion for systems with hundreds to millions of biochemical molecules like proteins, lipids and nucleic acids with complicated bonded interactions and LAMMPS [28], a classical molecular dynamics code that can model atomic, polymeric, biological, metallic, granular, and coarse-grained systems using a variety of force fields and boundary conditions.

Of particular note is the Green's Function Reaction Dynamics (GFRD) algorithm [29], based on the decomposition of the many-body problem of a reaction-diffusion system into one- and two-body problems that can be solved analytically via Green's functions. These Green's functions are then used to set up an event-driven algorithm which can be up to 5 orders of magnitude faster than using Brownian Dynamics, while modeling the stochastic character of the interactions among particles.

Despite the numerous capabilities and high efficiency of these simulators, they have not been designed with the application of DMC in mind. As a consequence, they are not able to compute fundamental magnitudes from a communication perspective such as signal amplitude, delay, throughput or noise. Therefore, some authors have developed new simulation frameworks specifically oriented for DMC. Several early molecular communication simulators are restricted to very simple scenarios where the transmitter releases a single particle [30, 31]. Moreover, diffusion is only modeled by means of Brownian motion (i.e., only normal diffusion is considered).

Funnycells [32] simulates populations of infected and antibody cells which compete with each other to infect/immunize as many cells as possible by releasing molecules into the environment. The medium is assumed to be sufficiently viscous so that the molecules stay immobile; the diffusion process is not implemented.

NanoNS [33] is an interesting simulation framework which has been built as a module of the well-known network simulator ns-2. This is a useful approach in order to facilitate the future study of the higher layers of a molec-

ular communication network. However, some aspects of NanoNS restrict its applicability to a general case of molecular communication. First, it is based on Fick’s laws of diffusion to determine the flux of particles between adjacent volume cells. This is a useful abstraction of the diffusion process, but it does not allow the simulation of anomalous diffusion or other effects, such as the noise due to Brownian motion [23]. Second, NanoNS models the diffusion process and the reception process with a single equation in the simulator algorithm, which makes it difficult to analyze separately these two key processes in a molecular communication system.

Another interesting simulator [34] models a generic set of nanomachines in a given environment. Communication may occur by means of a number of techniques, such as electromagnetic waves or calcium ions. Several configuration parameters allow to recreate different types of biological environments. The mobility model of both nanomachines and information carriers can also be personalized.

N3Sim (available for download at www.n3cat.upc.edu/n3sim) is the first open-source simulation framework for DMC which includes both the cases of normal and anomalous diffusion. In our previous work [35, 36], we provide a high-level overview of an initial version of *N3Sim* and describe its basic features. Since then, it has been extended by the addition of molecular harvesting and resource-constrained DMC [25]. Moreover, *N3Sim* has allowed the identification of important properties of DMC [37], the evaluation of modulation techniques and the validation of a physical channel model for DMC [38]. In this paper, we extend our previous work by providing a detailed description of the design choices and the implementation of some aspects of the latest version of *N3Sim*. Furthermore, simulation results in several simple scenarios are presented to illustrate the capabilities of the simulator.

4. Simulator Architecture

We designed *N3Sim* in order to simulate a set of nanomachines which communicate among them through molecular diffusion in a fluid medium [22]. The information to be sent by the transmitter nanomachines modulates the rate at which they release molecules, modeled as particles (see Section 4.2), to the medium. For instance, the transmission of a logical bit ‘1’ may be represented by the emission of a set of particles, and the transmission of a bit ‘0’ by the absence of emission. These emissions create variations in the local concentration of particles, which propagate throughout the medium due

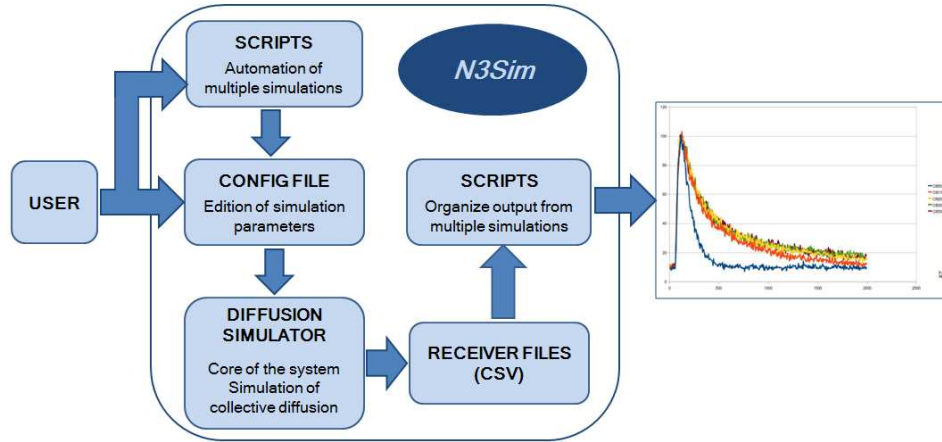


Figure 1: Block diagram of *N3Sim*

to the Brownian motion and to interactions among themselves. The receivers are able to estimate the concentration of particles in their neighborhood by counting the number of particles in a volume around their location. From this measurement, they can decode the transmitted information.

Figure 1 shows a block diagram of the steps needed to run a simulation. First, the user specifies the values of the simulation parameters in a configuration file. These parameters include the number and location of transmitters and receivers, the signal to be transmitted, the size of the emitted particles and the diffusion coefficient of the medium, amongst others. A script file allows the user to run multiple simulations automatically using only one configuration file, which is useful to easily evaluate the influence of a specific parameter (e.g., the number of transmitted particles) in the system output. Next, the diffusion simulator takes the configuration file and the automation scripts as input, and performs the actual simulation of the DMC scenario. The diffusion simulator computes the position and velocity of each particle in every time step of the simulation. When the simulation ends, its outputs are stored in receiver files (one per receiver), which contain the concentration measured by each receiver as a function of time. Last, another set of scripts may be used to organize the results from several receivers and graphically represent them into a single plot.

4.1. Node Models

A basic structure in the simulation framework is the *node*. A node is a generic agent which can be specialized into different categories depending on its function within the system. *N3Sim* currently implements three basic node types: transmitters, receivers and harvesters.

The transmitter is defined by its location in the simulation space and its size, which determines its influence space (i.e., the region where it may release particles). Every transmitter modulates the information to be sent into an associated waveform, which defines its particle release pattern. This waveform may be chosen among a number of predefined waveforms, such as a square pulse, a Gaussian pulse or a pulse train. A custom waveform may be defined by the user as well.

The receiver can be modeled as a sphere or cube able to measure the instantaneous number of particles within its detection range, from which the local concentration can be estimated. This model is an idealization of the ligand-receptor binding mechanism found in nature [39]. Many receiver parameters are customizable; for instance, after the receiver has detected a group of particles, it can either absorb them or be completely transparent to the particles.

N3Sim also allows the user to place multiple transmitters and receivers in the simulation field, thereby simulating a multipoint-to-multipoint communication. The user just needs to define the characteristics of each transmitter and receiver, namely, its position, shape and size, which can be different for each of them. This allows the study of several communication aspects, such as the feasibility of broadcast molecular communication, or the nature of interferences when more than one transmitter emits at the same time.

Finally, the third type of nodes in *N3Sim* is the harvester node. Harvesters have the ability of collecting molecules from their local neighborhood. These molecules are then stored into a *molecule reservoir* with a fixed capacity, and they may be used in future transmissions. Three main harvesting methods have been implemented in *N3Sim*:

1. *Standard harvesting* is the basic method, in which a node harvests molecules from the environment at the highest possible rate until the molecule reservoir is full. The reservoir is emptied after an emission of molecules, after which the harvesting operation can resume.
2. *Rate-limited harvesting* requires to specify a harvest rate up to which a node can harvest particles within its reception area. For instance, if a

harvester is limited to a 80% rate, only 80% of the molecules present in the neighborhood of the harvester will be collected at each time. This allows controlling the rate at which molecules in the environment will be absorbed due to the harvesting operation.

3. A node using the *unlimited harvesting* method is assumed to have an unlimited reservoir and to harvest every particle within its reception area. Therefore, *unlimited harvesting* can be considered as a special case of the *rate-limited harvesting* with an harvesting rate of 100%.

4.2. Particle Model

The emitted particles are modeled as indivisible spheres. The main reason for this choice is the simplicity of the collision detection algorithm (see 5) for this shape. Collisions between the emitted particles and the fluid molecules cause them to diffuse throughout the medium. It would be computationally infeasible to model each of these collisions individually, since the number of collisions between each particle and the fluid molecules is in the order of 10^{20} per second [40]. Fortunately, the seemingly random movement of the suspended particles caused by collisions between the particles and the smaller fluid molecules can be mathematically modeled as Brownian motion. Assuming that the particles have no inertia, Brownian motion allows to statistically calculate their movement, which can be modeled as a Gaussian random variable with zero mean and whose root mean square displacement in each dimension after a time t is $\sqrt{2Dt}$, where D is the diffusion coefficient of the medium [41].

The high-level effect of the particles moving with a Brownian pattern is their diffusion throughout the medium, according to Fick's laws of diffusion [17]. However, Fick's laws do not take into account the influence of interactions among the particles themselves (such as collisions and electrostatic forces), and thus they are only valid when the particle concentration is very low and these interactions are neglected.

In order to simulate environments with anomalous diffusion, which cannot be modeled by Fick's laws of diffusion, *N3Sim* is able to account for the inertia of the emitted particles and the interactions among them. The most relevant interactions which affect the particle diffusion process are collisions and electrostatic forces. For simplicity, collisions among particles are considered elastic (i.e., the total kinetic energy is conserved). Electrostatic forces appear when the emitted particles are ions, which have a non-zero electric charge.

4.3. Simulation Space

N3Sim can be configured to simulate either a 2-dimensional or a 3-dimensional space. One of the parameters of the simulation space is the initial particle concentration in the medium. If it is zero, both a bounded and an unbounded space can be simulated. However, if the initial particle concentration is greater than zero, the simulation space needs to be bounded, in order to avoid having an infinite number of particles.

When a bounded space is simulated, a cuboidal simulation space is assumed where particles rebound on the space limits. We consider this model to be the most realistic for prospective applications of molecular communication. For instance, in a set of communicating nanomachines located in a blood vessel, whenever a particle collides with the vessel wall (known as *tunica intima*), it will rebound. In order to avoid that the particles released by the transmitters cause the background concentration to increase over time, *N3Sim* includes an optional mechanism which lets some particles disappear when they reach the limits of the simulation space, according to the laws of diffusion.

Moreover, objects can also be set within the scenario in order to simulate obstacles between the transmitters and the receivers. This allows recreating, for example, a scenario where a group of bacteria is crossing the medium and it obstructs the way between transmitters and receivers.

Table 1 outlines some of the most relevant parameters that need to be configured to configure a simulation with *N3Sim*. A complete list of all the simulation parameters and their detailed description can be found in the website <http://www.n3cat.upc.edu/tools/n3sim/ParameterList>

5. Collision Detection

The problem of detecting collisions between n bodies with random movement is well-known and widely studied in the algorithmic domain. Due to their high computational cost, these algorithms represent a bottleneck in most of the applications that have to implement it.

Besides dealing with the n -body interaction, which is in itself a problem with a computational cost of $O(n^2)$, the problem of collision detection has an additional requirement. Collisions must be solved sequentially in time, since each collision changes the trajectory of the particles involved and may cause new potential collisions. In consequence, the collision detection process is mostly sequential. Since it is not possible to find a collision until the previous

Parameter name	Units	Description
time	ns	Total time of the simulation
timeStep	ns	Duration of each time step
boundedSpace	true/false	A rectangular bounded / unbounded space is simulated
xSize	nm	Horizontal size of the simulation space
ySize	nm	Vertical size of the simulation space
bgConcentration	particles/10000 nm ²	Initial background concentration
emitters	integer	Number of emitters in the simulation
emitterRadius	nm	Radius of the influence area of the emitter
emitterType	integer (1 to 5)	Emitter type (determines the emission pattern)
receivers	integer	Number of receivers in the simulation
emitterRadius	nm	Radius of the influence area of the emitter
emitterType	integer (1 to 3)	Receiver type (determines the detection area shape)

Table 1: Main parameters of *N3Sim*

one has been solved, it is difficult to parallelize any significant part of the algorithm.

5.1. State of the Art

A naive solution for the collision detection problem is to iterate through all pairs of objects, testing possible collisions, choosing the first one, solving it, move all particles to the time of this collision and repeat the process until end time is reached. The time complexity of the algorithm is $O(n_c * n^2)$, where n_c is the number of collisions and n is the number of objects. This high computation cost would cause the collision detection algorithm to become a bottleneck for the application performance.

There are algorithms that try to reduce this the computational cost of collision detection by leveraging the temporal and spatial coherence of collisions. Spatial coherence is the property that objects are more likely to collide with other objects in their neighborhood; as a consequence, the number of colliding object pairs at any time is much smaller than the total number of pairs. Temporal coherence means that the particles positions changes relatively little over small time intervals; therefore, calculations from previous frames can be reused to avoid unnecessary recomputation.

A well-known collision detection algorithm is *Baraff's algorithm*, also known as *sweep and prune* [42], which detects collisions *a posteriori*, i.e.,

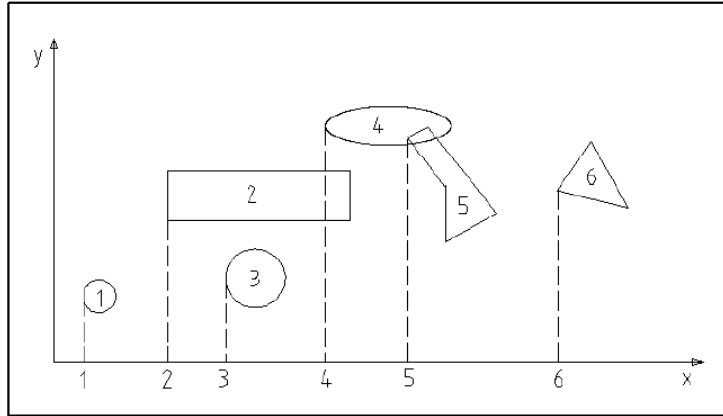


Figure 2: Schematic description of Baraff’s algorithm for collision detection. The position of an object is given by its lowest x coordinate.

after they have occurred. Baraff’s algorithm takes advantage of spatial coherence as follows. For each frame, the algorithm first sorts the objects by their lowest x coordinate, as shown in Fig. 2. Then, performing an appropriate iteration over the objects list (each object must be checked with all the others), most of the pairs of objects can be easily dismissed. As it can be seen in Fig. 2, for two objects to overlap (which means they have collided) it is necessary (but not sufficient) that their projections on each axis overlap. When looking for collisions, each object is compared only with the following ones in the sorted list. It is easy to see that, if the current object we are comparing does not overlap with a given object, it will not overlap with any of the following objects in the sorted list. This way, a large number of object pairs can be ruled out a priori, saving a significant time.

Baraff’s algorithm also takes advantage of the temporal coherence in the following way. After solving the collisions of a frame, the objects list must be sorted again, since objects may have changed their positions. Since the position changes are small, the list will be almost sorted and the sorting algorithm will be more efficient.

5.2. Algorithm implemented in *N3Sim*

The algorithm implemented in *N3Sim* is a variation of Baraff algorithm, which allows using it as an *a priori* algorithm to ensure that no collisions will be missed. The modification of Baraff’s algorithm has two parts. First,

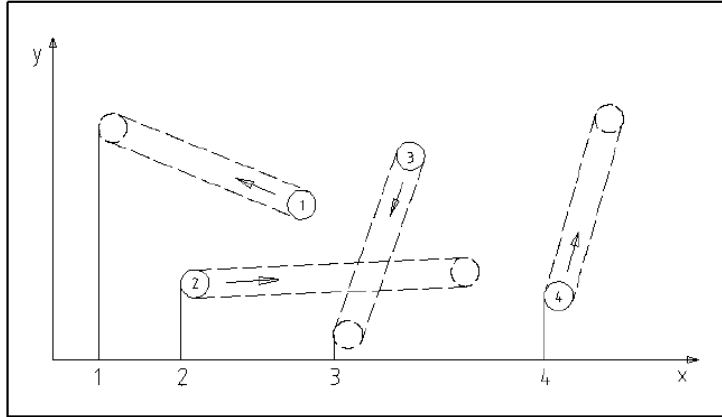


Figure 3: Schematic description of the algorithm used for collision detection in *N3Sim*. Objects are sorted by the minimum x coordinate of their trajectories.

instead of searching the overlap between two objects, the overlap between their trajectories will be considered. Second, it will be necessary to save all possible collisions found, sorted by time, into a data structure in order to take advantage of the temporal coherence.

Fig. 3 shows that for two particles to collide it is necessary (but not sufficient) that the projections of their trajectories overlap on each axis. It follows the same reasoning than Baraff's algorithm, substituting objects by their trajectories.

Following this analogy, the first step is to sort the objects list by the minimum x dimension of their trajectories. As in Baraff's algorithm, each object is tested against the next ones in the list, discarding all of the following when the trajectories of two objects do not overlap. Collisions are stored in a data structure known as collisions queue, sorted by collision time. This way, the cost to obtain the next collision is always $O(1)$.

Once the collisions queue has been created, the algorithm consists in obtaining the next collision and solving it until the queue is empty. Once a collision is solved, one must consider that the trajectories of the objects involved in the collision will have changed. As a consequence, two effects need to be taken into account: first, the objects list will no longer be sorted, and second, the previous collisions involving the objects of the recently solved collision are no longer valid.

To solve the first problem, it is evident that the objects list must be

sorted again. As in Baraff’s algorithm, an insertion sort algorithm is used because the list is almost sorted (just two objects are unsorted and they are almost at the right position in the list). Regarding the second problem, all collisions involving the objects of the solved collision must be deleted from the collisions queue. Searching through the collisions queue to delete such collisions would have a high time cost; instead, a tag system is used. Each object has a tag, an integer that is incremented each time the object is in a collision that is solved. On the other hand, each collision has two tags, one per object. These tags take the value of the object tag when the collision is created. Then, when a collision is retrieved from the collision queue to be solved, if the collision tags and the object tag do not match, it means that these objects have been in a previous collision; then, the current collision must be discarded and the next one must be solved. Following this procedure, the cost of discarding collisions is $O(1)$.

Finding the new collisions for the objects involved in a collision means to compare the object with the rest of the objects in the sorted list (as it is done in the first part of the algorithm) and to save the new collisions into the collisions queue. The cost of this step is $O(n \log(n))$, $O(n)$ to compare the object with the object in the list and $O(\log(n))$ to save each collision in the collisions queue.

Finally the data structures that store collisions have been modified to improve the memory costs. The problem that leads to these changes was that in simulations of large time steps and high number of collisions there were too many false collisions (collisions with invalid tags as explained in the previous section). These produced that the collisions queue grew too much. To avoid this problem, a list of collisions is associated with each object. This list is deleted and recalculated when a particle has a collision. This way, the memory cost is kept constant.

5.3. Cost Analysis

This collision detection algorithm implemented in *N3Sim* is summarized in the following table, where n is the number of objects and n_c is the number of collisions:

Stage	Operation	Time cost
1	Pre-processing	
1a	Sort objects list	$O(n)$
1b	Create collisions queue	$O(n^2 \log(n))$
2	Processing	
2	While collisions queue is not empty	n_c times
2a	Obtain first collision	$O(1)$
2b	Solve collision	$O(1)$
2c	Sort objects list	$O(n)$
2d	Delete invalid collisions	$O(1)$
2e	Find new collisions	$O(n \log(n))$

We conclude that the total computation cost of the collision detection algorithm for each time step is:

$$C = O(n^2 \log(n)) + n_c O(n \log(n)) \quad (4)$$

Similarly to the original Baraff's algorithm, there is a high pre-processing cost of $O(n^2 \log(n))$, but then the cost of detecting each collision is reduced from $O(n^2)$ (for a naive algorithm) to $O(n \log(n))$.

The number of collisions n_c is approximately:

$$n_c \approx O(n \cdot c \cdot t) \quad (5)$$

where c is the concentration of particles and t is the simulation time. If we consider a fixed space, the particle concentration is proportional to the number of particles, then

$$n_c \approx O(n \cdot c \cdot t) = O\left(n \cdot \frac{n}{area} \cdot t\right) = O(n^2 \cdot t) \quad (6)$$

and

$$C = O(n^2 \log(n)) + n_c O(n \log(n)) = O(n^2 \log(n)) + t \cdot O(n^3 \log(n)) \quad (7)$$

The memory cost depends on the number of collisions. As outlined earlier, the number of collisions depends linearly on the number of particles, concentration and time. The relevant time in terms of memory is a time step t_s , which is the time during which the algorithm will store the collisions. Therefore, we have the following expression for the memory cost in *N3Sim*:

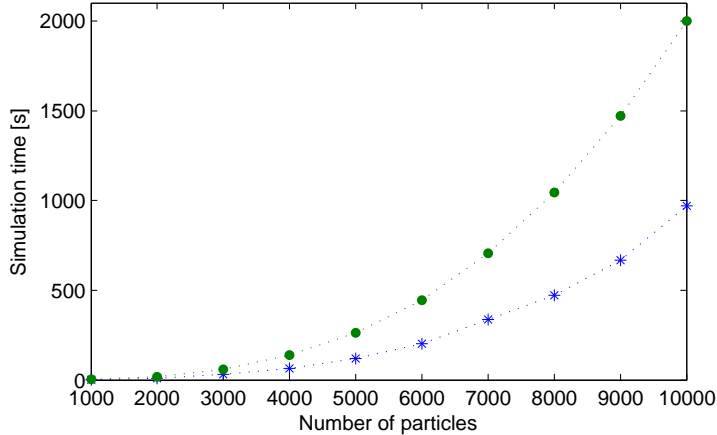


Figure 4: Comparison of the computational cost of the collision detection algorithm used in *N3Sim* (blue stars) and a brute force approach (green dots) as a function of the number of particles.

$$C_{mem} = O(n \cdot c \cdot t_s) = O(n^2 \cdot t_s) \quad (8)$$

5.4. Cost evaluation

We describe next some sample simulations performed to validate the previously calculated expressions of the scalability of collision detection as a function of the number of particles. In particular, the computational cost of the collision detection algorithm used in *N3Sim* (explained in Section 5.2) is compared to a naive algorithm following a brute force approach. We define a closed simulation space with a squared shape and dimensions 100x100 nm, which contains a number of particles ranging from 1000 to 10000, in intervals of 1000 particles. The simulation time is 10 μ s. All tests have been performed on an Intel i7-920 CPU with 8 GB of RAM running Java (OpenJDK) version 1.7.0_25 on a Linux kernel version 3.8.0-29. Figure 4 shows a comparison of the total duration of the simulations as a function of the number of particles present in the environment. As expected, the *N3Sim* collision detection algorithm (blue stars) shows a significant improvement in the simulation time with respect to the naive algorithm (green dots). Moreover, Figure 4 confirms the scalability advantage of the *N3Sim* algorithm, which improves the cost to detect a collision from $O(n^2)$ (for a naive algorithm) to $O(n \log(n))$.

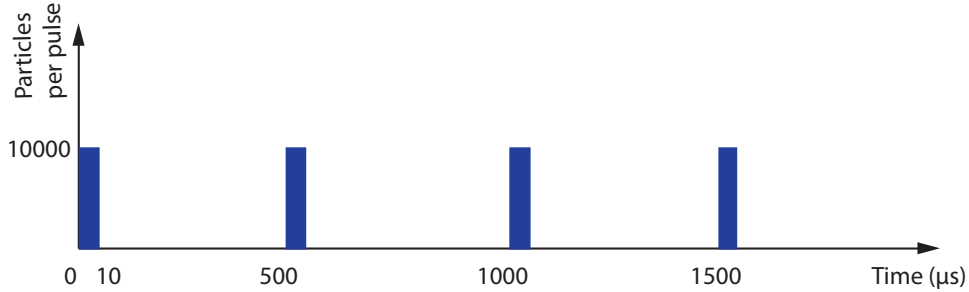


Figure 5: Train of square shaped pulses released by the transmitter node.

6. Simulation Results

As previously mentioned, *N3Sim* can simulate a broad range of scenarios, from very simple ones, consisting of a single receiver and a transmitter which releases defined patterns of particles, to more complex ones containing multiple nodes with particle harvesting capabilities and with different particle release patterns. These scenarios can be run inside a bounded or an unbounded space, with or without a background concentration of particles.

As a first simple example, Fig. 5 represents the transmitted signal by a single transmitter, which consists of a train of square pulses released every 500 μs . Each pulse contains 10000 particles and has a duration of 10 μs ; therefore, the emission of particles is almost instantaneous with respect to the interval between pulses. A receiver, located 500 nm away from the transmitter, measures the number of particles within its reception area. Both the transmitter and receiver nodes have a radius of 100 nm, and the simulation scenario is unbounded.

The received signal is shown in Fig. 6, which shows that the receiver measures a rapid increase on the particle concentration at intervals matching the transmission of pulses. It can also be observed that the particles do not disappear after they are measured by the receiver, but they remain in the environment and constitute a source of molecular noise [23].

The implementation of nodes with particle harvesting capabilities is an important feature of *N3Sim*. With their internal reservoirs, harvesting nodes can accumulate particles for future usage. Fig. 7 represents the consecutive charge and drain operations of the particle reservoir of a harvesting node. In this scenario, the harvesting node is continuously capturing particles from the environment during the execution of the simulation using the standard harvesting method. A pulse containing 1000 particles is released whenever

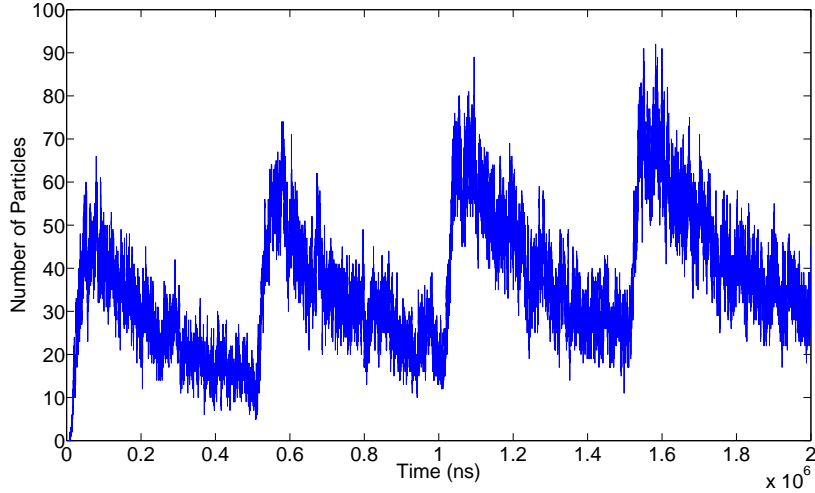


Figure 6: Measured signal by the receiver node.

the reservoir has enough particles. We can observe in Fig. 7 that the first harvesting operation (corresponding to the first peak) requires a longer charge time than the consecutive harvesting operations. The reason for this behavior is that the following harvesting operations are able to collect remaining particles from the previous pulse transmissions, thereby achieving a higher harvesting rate.

The impact of harvesting operation over the particle reception performance of a distant receiver is an interesting subject which can be easily evaluated using *N3sim*. A basic setup containing a transmitter, a receiver and a harvesting node, as shown in Fig. 8, can be utilized for this purpose. A two-dimensional bounded space is used with dimensions of 5000 nm by 5000 nm. The center-to-center distance between the transmitter and harvester node is 400 nm. Similarly, the distance between the receiver and harvester nodes is also 400 nm. The radius of all three nodes is 100 nm.

In this scenario, we can measure how the harvesting operation affects the signal at the receiver location by observing the maximum number of particles measured by the receiver during the simulation, after a pulse with a given amplitude is transmitted. This metric is illustrated in Fig. 9, which evaluates the reception performance of the receiver when the transmitter emits a pulse with an amplitude ranging from 25000 particles (bottom line) to 100000 particles (top line), in intervals of 25000 particles. The harvesting node

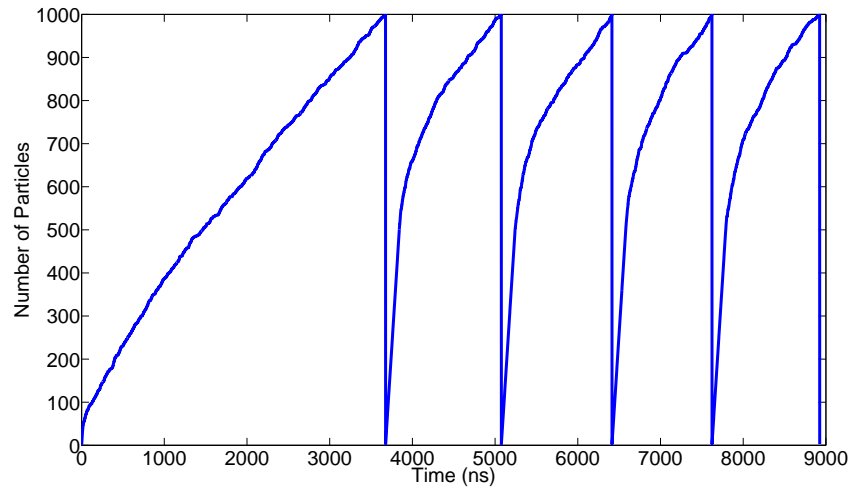


Figure 7: Charge/drain operations of the reservoir of a receiver

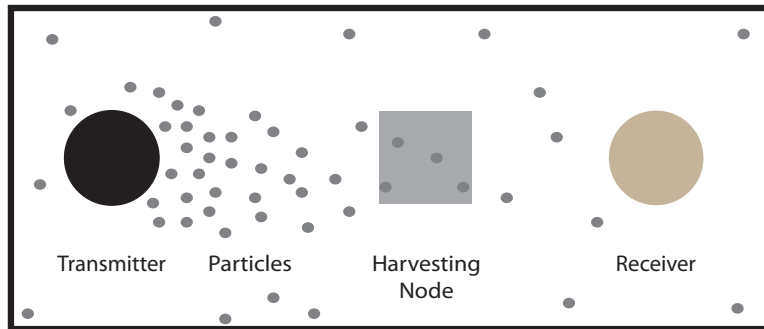


Figure 8: Setup containing a transmitter, a receiver and an harvesting node.

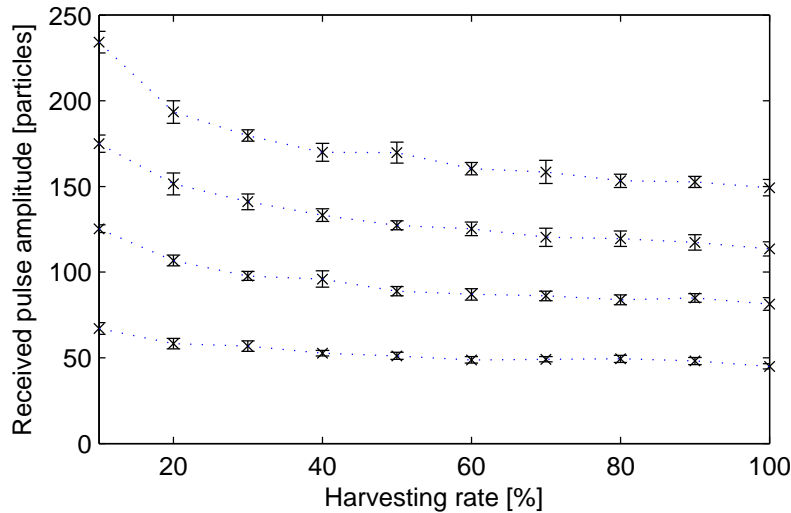


Figure 9: Amplitude of the signal measured by the receiver, when a pulse with an amplitude ranging from 25000 particles (bottom line) to 100000 particles (top line) is transmitted, as a function of the harvesting rate.

collects particles from the environment at a rate ranging from 10% to 100%, in 10% intervals. The averaged results of 10 simulation runs, along with the 95% confidence intervals, are shown. We observe that, independently of the amplitude of the transmitted pulse, increasing the harvesting rate causes the received signal to decrease. This decrease, due to the absorption of part of the transmitted signal by the harvesting node, is particularly significant when the harvesting rate changes from 10% to 40%, after which the amplitude of the received signal remains relatively constant. In any case, we observe a trade-off between the received signal and the harvesting rate; higher pulse amplitudes along with low harvesting rates result in the highest signal amplitude at the receiver location.

Fig. 10 shows the same experiment using an *unlimited harvesting* node. Therefore, the observed results are similar with those in the previous simulation for an harvesting rate of 100%. As expected, the signal measured by the receiver increases almost linearly with respect to the amplitude of the transmitted pulses.

We evaluate next two important metrics that can be also observed using *N3Sim*: the *maximum pulse frequency* and the *maximum achievable throughput*. The considered scenario in this case is slightly different than that of the

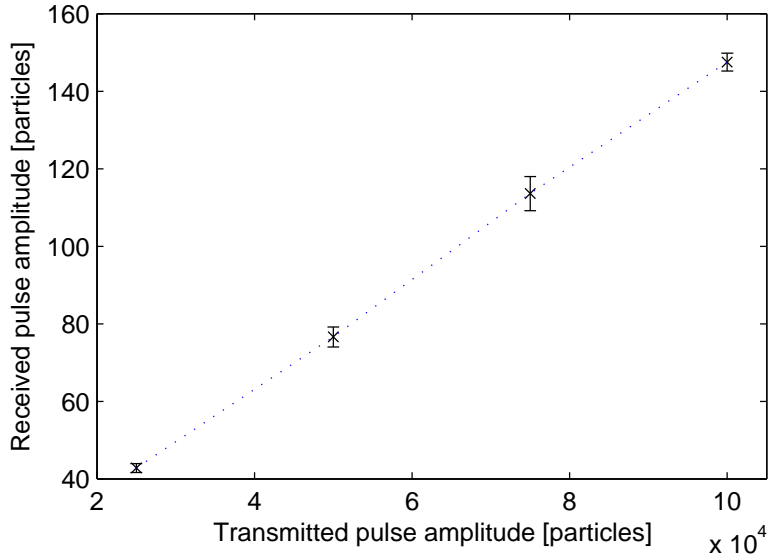


Figure 10: Amplitude of the signal measured by the receiver, as a function of the transmitted pulse amplitude, when using an unlimited harvesting node.

previous simulations. In particular, the transmitter node is enhanced with standard harvesting capabilities, so there are two nodes with harvesting capabilities in total. There exists an initial background concentration of particles homogeneously distributed over the bounded simulation space. Having no particles inside its reservoir at the beginning, the transmitting node continuously harvests particles, and emits a pulse of 1000 particles as soon as its reservoir charge reaches this level. Analogously to the scenario depicted in Fig. 8, a harvesting node, using the rate-limited harvesting method, is located between the transmitter and receiver nodes.

In this context, the *maximum pulse frequency* is defined as the number of pulses that the transmitter node can emit per unit of time, which is mainly limited by the rate at which the transmitter is able to harvest particles from the environment. The *maximum achievable throughput* is defined as the number of pulses that are successfully detected by the receiver node per unit of time. The receiver decodes the received pulses by means of amplitude detection [43]; i.e., a pulse is correctly detected if the maximum particle concentration measured by the receiver exceeds a given threshold. The main difference between these two metrics is that, in some cases, not

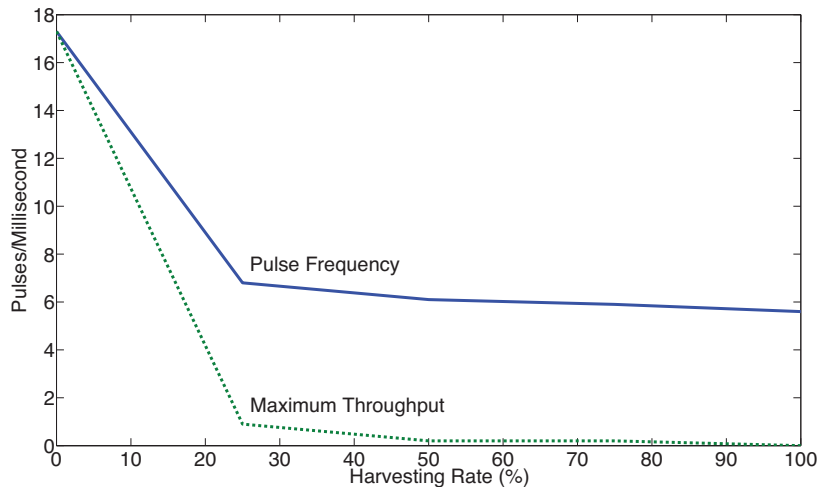


Figure 11: Maximum pulse frequency at the transmitter (blue solid line) and maximum achievable throughput at the receiver (green dashed line), for different harvesting rates.

all transmitted pulses will be correctly detected by the receiver, due to the random diffusion noise or to the absorption of particles by the harvesting node. Therefore, the maximum achievable throughput will always be lower or equal to the maximum pulse frequency.

Fig. 11 shows the maximum pulse frequency of the transmitter (blue solid line) and the maximum achievable throughput as measured by the receiver (green dashed line), as a function of the harvesting rate. We observe that the harvesting operation has a devastating impact on both of these metrics, as they sharply decrease when the harvesting rate changes from 0% to 25%. We conclude that the harvester node should have a low harvesting rate (below 25%) in order to allow for a successful communication in the considered scenario.

7. Conclusions and Future Work

In this work, we have analyzed diffusion-based molecular communication, a promising bio-inspired paradigm to implement nanonetworks, i.e., the interconnection of nanomachines. The physical channel of diffusion-based molecular communication is fundamentally different from that of classical electromagnetic communication; as a consequence, nanonetworks require novel

physical channel models, network architectures and communication protocols, which need to be validated by simulation.

With this purpose, we have presented *N3Sim*, a simulation framework for diffusion-based molecular communication, which we consider an essential tool to design and evaluate physical-layer protocols, modulations, resource management schemes and nanomachine components, amongst others. *N3Sim* allows to simulate scenarios where transmitters encode the information by releasing particles into the medium, thus causing a variation in their local concentration. *N3Sim* models the movement of these particles as Brownian motion, and it also takes into account their inertia and the interactions among them. Receivers decode the information by sensing the particle concentration in their neighborhood.

We expect that *N3Sim* will play an important role to evaluate current models, modulations and protocols for diffusion-based molecular communication, and that it will give crucial insights which may help to design novel techniques and schemes.

N3Sim does not scale well to large scenarios with a high background concentration, since it needs to compute the motion and the interactions among every single particle in the simulation space. Therefore, in order to design higher-level protocols (such as MAC or routing schemes) for complex scenarios, there may be the need for a higher-level simulator (which could be integrated into a network simulator, such as ns-3) with a better scalability at the cost of precision. This simulator would not account for every individual particle, but for the distribution of the particle concentration throughout time and space, in a scenario of either normal or anomalous diffusion. A comparison of the computational efficiency of *N3Sim* with respect to state-of-the-art simulators would also be useful.

Another future research line considers modifying *N3Sim* in order to simulate other scenarios of molecular communication. Examples include medium-range communications based on flagellated bacteria [9], which move following the concentration of attractants diffused in the medium by nanomachines, and taking into account the electrostatic forces among particles.

Acknowledgment

This work has been partially supported by the FPU grant of the Spanish Ministry of Education and by the Science Fellowships and Grant Programmes Department (BIDEB) of The Scientific and Technological Research Council

of Turkey (TUBITAK). The authors would also like to thank Iñaki Pascual for his help with the implementation of *N3Sim*.

References

- [1] K. E. Drexler, *Nanosystems: Molecular machinery, manufacturing, and computation*, Wiley, 1992.
- [2] R. A. Freitas, *Nanotechnology, nanomedicine and nanosurgery.*, *International Journal of Surgery* 3 (2005) 243–6.
- [3] C. M. J. Pieterse, M. Dicke, *Plant interactions with microbes and insects: from molecular mechanisms to ecology*, *Trends in plant science* 12 (2007) 564–9.
- [4] J. Han, J. Fu, R. B. Schoch, *Molecular sieving using nanofilters: past, present and future*, *Lab on a chip* 8 (2008) 23–33.
- [5] I. F. Akyildiz, J. M. Jornet, *Electromagnetic wireless nanosensor networks*, *Nano Communication Networks* 1 (2010) 3–19.
- [6] S. Abadal, E. Alarcón, M. C. Lemme, M. Nemirovsky, A. Cabellos-Aparicio, *Graphene-enabled Wireless Communication for Massive Multicore Architectures*, *IEEE Communications Magazine* (2012).
- [7] I. F. Akyildiz, F. Brunetti, C. Blázquez, *Nanonetworks: A new communication paradigm*, *Computer Networks* vol (2008) 52no12pp2260–2279.
- [8] J. M. Jornet, I. F. Akyildiz, *Graphene-Based Nano-Antennas for Electromagnetic Nanocommunications in the Terahertz Band*, in: *European Conference on Antennas and Propagation*, Barcelona.
- [9] M. Gregori, I. Llatser, A. Cabellos-Aparicio, E. Alarcón, *Physical channel characterization for medium-range nano-networks using flagellated bacteria*, *Computer Networks* (2010).
- [10] M. Gregori, I. Llatser, A. Cabellos-Aparicio, E. Alarcón, *Physical channel characterization for medium-range nanonetworks using catalytic nanomotors*, *Nano Communication Networks* 1 (2010) 102–107.
- [11] L. Parcerisa, I. F. Akyildiz, *Molecular communication options for long range nanonetworks*, *Computer Networks* 53 (2009) 2753–2766.

- [12] T. Nakano, T. Suda, M. J. Moore, R. Egashira, A. Enomoto, K. Ito, K. Arima, Molecular communication for nanomachines using intercellular calcium signaling, in: IEEE Conference on Nanotechnology, volume 2, pp. vol2478–481.
- [13] F. Walsh, S. Balasubramaniam, D. Botvich, W. Donnelly, Synthetic protocols for nano sensor transmitting platforms using enzyme and DNA based computing, *Nano Communication Networks* 1 (2010) 50–62.
- [14] D. Arifler, Capacity Analysis of a Diffusion-Based Short-Range Molecular Nano-Communication Channel, *Computer Networks* 55 (2011) 1426–1434.
- [15] L. Vlahos, H. Isliker, Y. Kominis, K. Hizanidis, Normal and anomalous diffusion: A tutorial, *Order and Chaos* 10 (2008) 1–40.
- [16] I. Karatzas, S. E. Shreve, *Brownian motion and stochastic calculus*, Springer, 1991.
- [17] J. Philibert, One and a Half Century of Diffusion: Fick, Einstein, before and beyond, *Diffusion Fundamentals* 4 (2006) 1–19.
- [18] D. E. Clapham, Calcium signaling, *Cell* 131 (2007) 1047–58.
- [19] A. Ladd, H. Gang, J. Zhu, D. Weitz, Time-dependent collective diffusion of colloidal particles., *Physical review letters* 74 (1995) 318–321.
- [20] J. Gillis, Correlated random walk, *Mathematical Proceedings of the Cambridge Philosophical Society* 51 (1955) 639–651.
- [21] A. Goldbeter, G. Dupont, M. J. Berridge, Minimal model for signal-induced Ca²⁺ oscillations and for their frequency encoding through protein phosphorylation, *Proc. of the National Academy of Sciences of the United States of America* 87 (1990) 1461–5.
- [22] M. Pierobon, I. F. Akyildiz, A physical end-to-end model for molecular communication in nanonetworks, *IEEE Journal on Selected Areas in Communications* 28 (2010) 602–611.
- [23] M. Pierobon, I. F. Akyildiz, Diffusion-based Noise Analysis for Molecular Communication in Nanonetworks, *IEEE Transactions on Signal Processing* 59 (2011) 4168–4182.

- [24] B. Atakan, O. B. Akan, On Channel Capacity and Error Compensation in Molecular Communication, Springer Trans. on Computational System Biology (2008) 59–80.
- [25] D. Demiray, A. Cabellos-Aparicio, E. Alarcón, D. T. Altilar, I. Llatser, DIRECT: A model for networks that are based on discrete entities to encode messages, submitted (2013).
- [26] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kal, K. Schulten, Scalable molecular dynamics with namd, Journal of Computational Chemistry 26 (2005) 1781–1802.
- [27] E. Lindahl, B. Hess, D. Van Der Spoel, Gromacs 3.0: a package for molecular simulation and trajectory analysis, Molecular modeling annual 7 (2001) 306–317.
- [28] S. Plimpton, P. Crozier, A. Thompson, Lammmps-large-scale atomic/molecular massively parallel simulator, Sandia National Laboratories (2007).
- [29] J. S. van Zon, P. R. Ten Wolde, Greens-function reaction dynamics: A particle-based approach for simulating biochemical networks in time and space, The Journal of chemical physics 123 (2005) 234910.
- [30] M. J. Moore, T. Suda, K. Oiwa, Molecular communication: modeling noise effects on information rate, IEEE Transactions on Nanobioscience 8 (2009) 169–80.
- [31] S. Kadloor, R. Adve, A Framework to Study the Molecular Communication System, in: International Conference on Computer Communications and Networks, San Francisco.
- [32] Funnycells, <http://www.cse.iitd.ernet.in/~aseth/assg/funnycells/funnycells.html>, 2010.
- [33] E. Gul, B. Atakan, O. B. Akan, NanoNS: A nanoscale network simulator framework for molecular communications, Nano Communication Networks 1 (2010) 138–156.
- [34] L. Felicetti, M. Femminella, G. Reali, A simulation tool for nanoscale biological networks, Nano Communication Networks 3 (2012) 2–18.

- [35] I. Llatser, I. n. Pascual, N. Garralda, A. Cabellos-Aparicio, E. Alarcón, N3Sim: A Simulation Framework for Diffusion-based Molecular Communication, *IEEE TC on Simulation* (2011) 3–4.
- [36] I. Llatser, N. Garralda, A. Cabellos-aparicio, M. Pierobon, E. Alarcón, J. Solé-Pareta, Exploring the Physical Channel of Diffusion-based Molecular Communication by Simulation, in: *IEEE GLOBECOM*, pp. 566–570.
- [37] N. Garralda, I. Llatser, A. Cabellos-Aparicio, M. Pierobon, Simulation-based Evaluation of the Diffusion-based Physical Channel in Molecular Nanonetworks, in: *IEEE International Workshop on Molecular and Nanoscale Communications (MoNaCom)*, pp. 443–448.
- [38] I. Llatser, E. Alarcón, M. Pierobon, Diffusion-based Channel Characterization in Molecular Nanonetworks, in: *IEEE International Workshop on Molecular and Nanoscale Communications (MoNaCom)*, pp. 467–472.
- [39] J. P. Rospars, V. Krivan, P. Lánský, Perireceptor and receptor events in olfaction. Comparison of concentration and flux detectors: a modeling study., *Chemical senses* 25 (2000) 293–311.
- [40] J. Piasecki, Centenary of Marian Smoluchowski’s Theory of Brownian Motion, *Acta Physica Polonica Series B* 38 (2007) 1623.
- [41] A. Einstein, *Investigations on the theory of the brownian movement*, 1915.
- [42] D. Baraff, *Dynamic Simulation of Non-Penetrating Rigid Bodies*, Ph.D. thesis, Computer Science Department, Cornell University, 1992.
- [43] I. Llatser, A. Cabellos-Aparicio, M. Pierobon, E. Alarcón, Detection Techniques for Diffusion-based Molecular Communication, to appear in *IEEE Journal on Selected Areas in Communications* (2012).