

# dMCS: Distributed Molecular Communication Simulator

Ali Akkaya\*  
Bogazici University  
Dept. of Computer Engineering  
Computer Networks Research Laboratory  
(NETLAB)  
34342, Istanbul, Turkey  
ali.akkaya@boun.edu.tr

Tuna Tugcu  
Bogazici University  
Dept. of Computer Engineering  
Computer Networks Research Laboratory  
(NETLAB)  
34342, Istanbul, Turkey  
tugcu@boun.edu.tr

## ABSTRACT

Nanonetworking is a new research field in which nanotechnology and communication engineering disciplines are employed to explore the possible communication mechanisms at nanoscale. Inspired by living organisms, molecular communication is one of the alternatives that can be used for communication between nanomachines. The research activities on molecular communication depend on simulations to verify and analyze the performance of proposed models. Due to the different channel characteristics, current simulation tools can not be used as is for nanonetworking. Simulation at nanoscale requires modeling of new communication paradigm, hence either existing tools need to be extended, or new tools need to be developed. Since molecular communication involves the modeling of large number of nanoscale objects, scalability of the simulation tool is another important concern. In this paper, we introduce dMCS, a distributed molecular communication simulator design. The proposed architecture is based on High Level Architecture (HLA), which is standardized under IEEE 1516. The results show that using the proposed architecture, it is possible to exploit different scalability options to shorten the execution time significantly. This enables modeling large and complex system simulations.

## 1. INTRODUCTION

Nanonetworking is a new research area, focusing on communication needs of nanoscale devices. The communication between nanomachines allows cooperation and collaboration among them in order to perform complex tasks. This enables application of nanotechnology and nanonetworking to different fields including biomedicine, industry, military and environment [4]. Molecular communication is one of

\*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 Scientific and Technical Research Council of Turkey (TUBITAK) under grant number 112E011.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

BODYNETS 2013, September 30-October 02, Boston, United States  
Copyright © 2013 ICST 978-1-936968-89-3  
DOI 10.4108/icst.bodynets.2013.253576

the methods that can be used for inter-nanomachine communication. Research on molecular communication heavily depends on simulation tools, either as the only means to analyze the proposed model, or to support the analytical model. A flexible and scalable simulation architecture is needed to increase the efficiency of all research activities. This work is an attempt to define such an architecture based on the standardized High Level Architecture.

Several research efforts have been focused on simulation of nanonetworking. In [10], the need for simulation is pointed out and the simulation requirements of molecular motor based communication network are defined very briefly. In [12], a simulator for 3D Brownian motion is proposed. The novelty of the proposed model is a dual time step approach to cope with the run time complexity of high number of particles. In [7], the authors introduce a simulation framework (NanoNS) developed on top of the commonly used NS-2 discrete event simulator. The authors use multi-particle lattice gas automata algorithm, in which the exact location of particles are not tracked but the medium is divided into lattice slides, to increase the efficiency of the tool. N3Sim [2] is a simulation tool for diffusion based molecular communication [9]. It enables the evaluation of performance of molecular networks and considers particle inertia and collisions among particles. In [6], a simulation platform for modeling communication at nano-scale is introduced. The model is defined to be generic and can be used for different communication options. A case study with molecular communication is used to demonstrate the simulator features.

In our paper, we introduce dMCS, a distributed molecular communication simulator. The design is based on HLA, which is standardized under IEEE 1516 [1]. We define the architecture and analyze the performance of the architecture for a use-case including multiple transmitter and receiver nanomachines. The simulation model can be used to simulate diffusion based molecular communication of in-body networks of engineered cells or artificially created nanomachines. In most of the previous works in the literature, simulation model is simplified with assumptions, which may affect the results, or only simple simulation scenarios are considered. HLA-based architecture enables horizontal scalability, which makes very large simulation scenarios possible.

## 2. BACKGROUND

### 2.1 Diffusion Based Molecular Communication Model

Nanomachines are the simplest devices in nanoscale that can accomplish basic computational tasks. In diffusion based molecular communication, a vesicle that contains a group of molecules is prepared before each symbol duration by transmitter nanomachine in encoding phase. At the start of the symbol duration, the molecules are released to the propagation medium. The molecules then diffuse through the propagation medium and arrive at the receiver nanomachine. Throughout the symbol duration, the molecules that reach the receptors of the receiver nanomachine are successfully received. The last step is the decoding of the received molecular signal to retrieve the encoded information.

## 2.2 High Level Architecture

HLA was developed by the United States Department of Defense (DoD) to integrate distributed simulation models within a common architecture. Since it has been accepted as common architecture for distributed simulation by the community, it was then standardized by the IEEE [1]. The main purpose of HLA is to define the component model and their interactions to create a collaborative system in which each simulation unit runs on an independent computational unit and communicate to simulate a scenario. The basic components in HLA are called federates, and they enable software reuse and abstraction. Run Time Infrastructure (RTI) is used for the communication of components to form a simulation model referred to as federation. The RTI is the backbone of the federation, and provides synchronization, communication, and data exchange services to the federates [3]. The HLA interface specification defines the way the services are accessed, but does not define how the services are actually implemented.

## 3. MOLECULAR COMMUNICATION SIMULATION ARCHITECTURE

Our proposed architecture design is shown in Figure 1. 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-based on common Federation Object Model (FOM). The scalability is achieved by distributing molecule related tasks to Molecule Federates, nanomachine related tasks to Nanomachine Federates, and slicing the 3D space and assigning a medium federate to manage each slice.

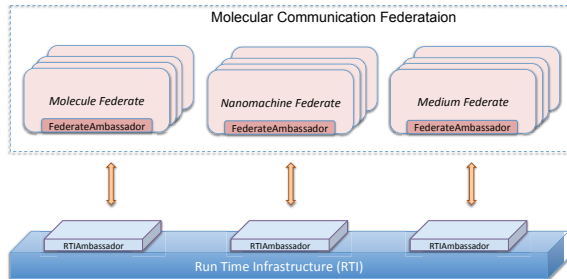


Figure 1: Simulation Architecture

### 3.1 Molecule Federate

The Molecule Federate abstracts the molecules in the simulation. It is responsible for molecule movements in the medium. It subscribes to the Medium Federate attributes,

meaning that in case the attributes of the Medium Federate changes, this information is communicated with the Molecule Federate by RTI. The Molecule Federate calculates the next position of the molecules and updates the attributes accordingly. Currently an implementation for Brownian motion [5] is available, but it is possible to implement different diffusion laws. The molecules propagate through 3D space, and this movement is modeled as three independent displacements, one for each dimension in 3D space. Total displacement  $\vec{r}$  in one time step can be found as [8],

$$\vec{r} = (\Delta x, \Delta y, \Delta z). \quad (1)$$

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

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

where  $\mu$  is taken as 0, and  $\sigma$  is calculated as

$$\sigma = \begin{cases} \sqrt{\frac{K_b \cdot T}{2 \cdot \pi \cdot \eta \cdot r_s}} \Delta t, & \text{if } s_{molecule} \gg s_{fluid} \\ \sqrt{\frac{K_b \cdot T}{3 \cdot \pi \cdot \eta \cdot r_s}} \Delta t, & \text{if } s_{molecule} \approx s_{fluid} \end{cases} \quad (3)$$

where  $K_b$  is the Boltzmann constant,  $T$  is the temperature of the environment,  $\eta$  is the viscosity of the fluid, and  $r_s$  defines the Stokes' radius of the propagating molecule [11]. These environmental coefficients are defined as part of medium slices, and once a molecule moves into a slice, it starts moving according to the new coefficients. The interfaces of the federate to other federates are shown in Figure 2.

### 3.2 Nanomachine Federate

The Nanomachine Federate models the nanomachines in the simulation. It is responsible for molecule transmission and reception logic. It is possible to define different types of transmitter and receiver nanomachines with different behaviors. Different implementations of a transmitter nanomachine can release molecules in bulk or sequentially, from a single point, or from different points. An implementation of a receiver nanomachine can receive molecules from all over its surface, or another can receive only via receptors distributed on its surface. The communication mechanism depends on the encoding and decoding mechanisms implemented by the transmitter and receiver nanomachines. Our current implementation is a single release point bulk transmitter, and reception over the whole surface. The interfaces of the federate to other federates are shown in Figure 2.

### 3.3 Medium Federate

The Medium Federate abstracts the medium slices. It is possible to define medium slices to construct a larger 3D space for simulation. The collision handling is performed by the Medium Federate via molecule rollback interactions in intermediate time steps. The simulation scalability is achieved by assigning different medium slices to different

medium federates. The interfaces of the federate to other federates are shown in Figure 2.

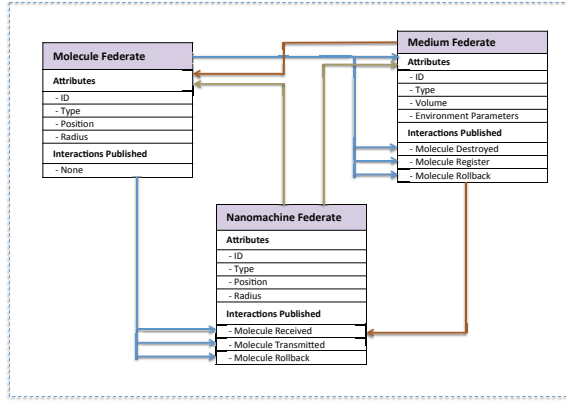


Figure 2: Federates

## 4. SIMULATION EXPERIMENTS

### 4.1 Simulation Scenario

A simulation scenario including four transmitter and four receiver nanomachines distributed in a 3D space is created using the proposed simulation tool. The simulation parameters are listed in Table 1. The molecules are transmitted by the transmitter nanomachines at the start of the symbol durations, each emitting one hundred molecules to represent binary *one* and zero to represent binary *zero*. The molecules diffuse from one medium slice to another if the medium is divided into slices, and if a molecule diffuses out through simulation boundaries, it is removed from the simulation.

Parameter	Value
Temperature	310 °K
Viscosity of the fluid	0.001 kg/s.m
Radius of the Transmitters	10 μm
Radius of the Receivers	10 μm
Radius of the Molecule	2.86 nm

Table 1: Simulation Parameters

### 4.2 Performance Evaluation

Since there is no standard benchmarking for comparison of runtime with other available simulators, we analyze the gain of the simulator for multi node architectures. For the performance evaluation of the distributed simulation, we define the speedup ( $S$ ) as

$$S = \frac{T_s}{T_m} \quad (4)$$

where  $T_s$  is the execution time with a single medium federate and  $T_m$  is the execution time with multiple medium federates, each managing a medium slice. The analysis is done with single, two, four, and eight medium federates. If there is a single federate, it manages all the simulation space. When there are two federates, the simulation space is sliced

into two, and each federate manages one slice. A similar logic is used for four and eight federates.

We run the simulation scenario for a collision free medium, where the molecules can coexist at the same position, and also with the collision management mechanism in which the molecule movements are evaluated in random order, and if the target location is occupied by another molecule, the movement is not performed. The results are depicted in Figure 3. For the collision free scenario, increasing the number of medium federates does not result in a speedup, while with the collision implementation, a speedup is achieved. Therefore, for the collision free scenario, the computational bottleneck is not in collision evaluation, and the gain of multiple nodes is lost on communication overhead among nodes. This shows that for such scenarios, exploiting the parallelism for molecule federate or nanomachine federate is needed. For collision implementation, when intensive computational work is distributed over multiple nodes, execution time decreases. As seen from Figure 3, the efficiency of the parallelism decreases after four nodes, since the communication overhead becomes dominant.

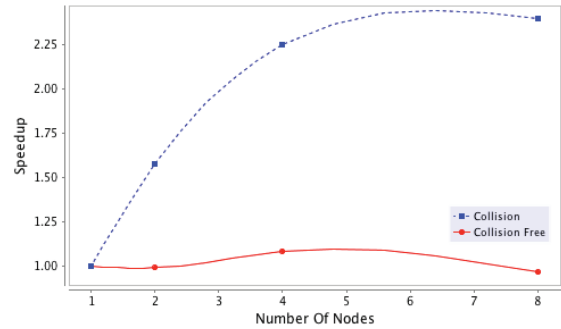


Figure 3: Collision and Collision Free

To analyze the effect of implemented collision handling algorithm on speedup, we implement dummy algorithms, which require 1 ms, 2 ms, 4 ms, 8 ms and 16 ms of computational time to execute per molecule movement. The results in Figure 4 show that, as the computational time for the medium federate increases, the system performance increases up to 8 ms. After that point, it does not get better towards the theoretical linear speedup limit due to the communication overhead. In the scenarios where a high computational load is required for collision management, the system achieves close to linear speedup.

For the cases where the bottleneck is not in collision management, different parallelism options need to be considered. This can be achieved by using multiple Nanomachine Federates, to manage a set of nanomachines. The scalability of the simulation will be dependent on the complexity of the algorithms used for transmission and reception processes. The simulation results for multiple Nanomachine Federates are shown in Figure 5. If a simple reception model is implemented in which whole surface is used for reception, adding more Nanomachine Federates does not decrease the execution time since the additional processing power can not be utilized due to networking overhead. Networking overhead becomes more dominant as the number of nodes increases.

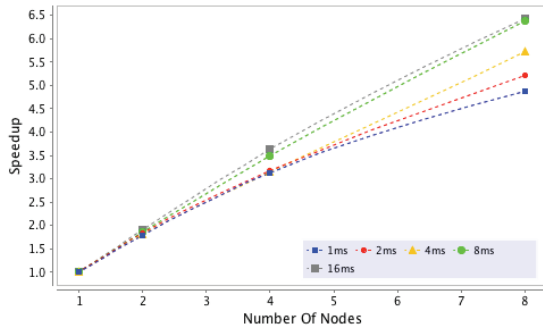


Figure 4: Effect of Algorithm Complexity

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

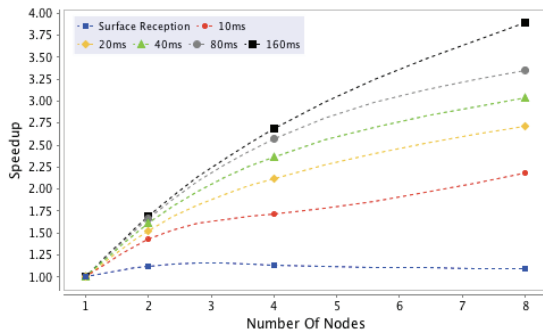


Figure 5: Nanomachine Federate Scalability

To analyze the simulation scenarios where the molecule movements model is complicated, we compare Brownian motion and more complex dummy algorithms, which require specific processing time. As shown in Figure 6, for Brownian motion, the network overhead is dominant and no gain is achieved in terms of execution time. When more complex algorithms are implemented, which take 1 ms, 2 ms, 4 ms, 8 ms, and 16 ms for molecule movement evaluation, better system speedup is observed. The results show that scalability of Molecule Federates need to be used for simulation scenarios where computationally intensive movement models need to be analyzed.

## 5. CONCLUSION

In this paper, we propose dMCS, a distributed simulation architecture for molecular communication. The architecture enables defining inter-operable and reusable simulation components, which can be executed in a distributed environment. This enables the execution of large scale and complex simulation scenarios. The architecture components are described and the performance of the simulation tool is demonstrated using a simulation scenario. It is shown that if computationally intensive collision algorithms, transmission or reception algorithms, and molecule movement algorithms need to be analyzed, the benefits of distributed architecture

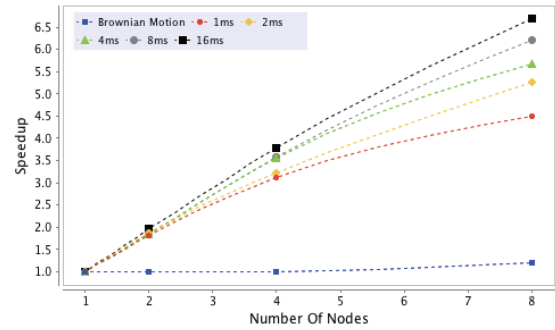


Figure 6: Molecule Federate Scalability

is fully utilized. As future work, we plan to extend the model to analyze different communication models in nanonetworks.

## 6. REFERENCES

- [1] Modeling and simulation (m & s) high level architecture (hla) (ieee 1516-2010 series).
- [2] N3sim: a simulation framework for diffusion-based molecular communication in nanonetworks, available at <http://www.n3cat.upc.edu/n3sim>.
- [3] S. AbouRizk, Y. Mohamed, H. Taghaddos, F. Saba, and S. Hague. Developing complex distributed simulation for industrial plant. In *Proceedings of the 2010 Winter Simulation Conference*, 2010.
- [4] I. F. Akyildiz, F. Brunetti, and C. Blázquez. Nanonetworks: A new communication paradigm. *Computer Networks*, 52(12), 2008.
- [5] S. F. Bush. *Nanoscale Communication Networks*. Artech House, 2010.
- [6] L. Felicetti, M. Femminella, and G. Reali. A simulation tool for nanoscale biological networks. *Nano Communication Networks*, 3(1), 2012.
- [7] E. Gul, B. Atakan, and O. B. Akan. Nanons: A nanoscale network simulator framework for molecular communications. *Nano Communication Networks*, 1(2), 2010.
- [8] M. Ş. Kuran, H. B. Yilmaz, T. Tugcu, and B. Özerman. Energy model for communication via diffusion in nanonetworks. *Nano Communication Networks*, 1(2), 2010.
- [9] I. Llatser, I. Pascual, N. Garralda, A. Cabellos-Aparicio, and E. Alarcon. N3sim: A simulation framework for diffusion-based molecular communication. In *IEEE Technical Committee on Simulation 8*, 2011.
- [10] M. Moore, A. Enomoto, T. Nakano, T. Suda, A. Kayasuga, H. Kojima, H. Sakakibara, and K. Oiwa. Simulation of a molecular motor based communication network. In *Bio-Inspired Models of Network, Information and Computing Systems, 2006. 1st*, 2006.
- [11] M. J. Saxton. *Modeling 2d and 3d diffusion, Methods in molecular biology*. Springer, 2007.
- [12] A. Toth, D. Banky, and V. Grolmusz. 3-d brownian motion simulator for high-sensitivity nanobiotechnological applications. *IEEE transactions on nanobioscience*, 10(4), 2011.