Simulating Wireless Nano Sensor Networks in the NS-3 platform

G. Piro, L.A. Grieco, G. Boggia, and P. Camarda
DEE - Dip. di Elettrotecnica ed Elettronica, Politecnico di Bari
v. Orabona, 4 - 70125, Bari, Italy
e-mail: {g.piro,a.grieco,g.boggia,camarda}@poliba.it

***Abstract*—The Wireless nanosensor network paradigm is rapidly gaining the attention of researchers within the scientific and industrial communities, thanks to the progress of nanotechnology. The envisioned concept is based on integrated machines at the nano scale, which interact on cooperative basis by means of wireless communications. At the present stage, the design of the protocol suite for wireless nanosensor networks represents a fundamental issue to address for accelerating the deployment process of such a technology. In this direction, the availability of an open source simulator would be surely of help in the development of participated design methodologies, in enforcing the collaboration among different teams, and in easing the verifiability of scientific outcomes. Despite the evident advantages that such a platform could bring, currently available tools only support molecular-based approaches without accounting for the relevant impact that electromagnetic communications may have in this field. To cover this lack, the present contribution proposes a modular and easy upgradeable simulation platform, intended for wireless nanosensor networks based on electromagnetic communication in the terahertz band. Preliminary results drawn from a a simple, yet significant, health monitoring scenario are also provided along with a study on computational requirements and future upgrades.**

***Keywords*-Nanonetworks, WNSN, Modeling, Simulation, Performance Evaluation**

I. Introduction

The progress of nanotechnologies is going to allow the development of nanomachines, i.e., integrated devices equipped with storage, processing, sensing, and communication units, with size ranging from one to few hundred of nanometers, very well suited for ICT, biomedical, multimedia, industrial, and military applications [1][2]. A nanodevice is expected to have very limited capabilities, so that it may only execute simple sensing, computing, actuation, and information storage tasks. This limits can be circumvented with more nanodevices cooperating in a collaborative fashion and communicating each other. In other words, the overall capability of many of them can be powerful enough to set-up a new network system, that is, a wireless nanosensor network (WNSN). This kind of cooperative network can be very useful for a lot of applications [1].

So far, different communication modes have been envisaged to let nanodevices communicating to each other: nanomechanical, acoustic, molecular, and based on electromagnetic waves. Among them, the one based on the electromagnetic (EM) radiation is rapidly gaining the attention of the scientific community. In fact, it is already possible to exploit some potentials of nanomaterials for performing, at the nanoscale, the transmission and the process of electromagnetic signals [3][4].

For the time being, preliminary research efforts have been devoted to characterize the wireless channel in the terahertz band at the nanoscale (see for example contributions provided in [5] and [6]).

Today research, starting from these precious outcomes, is considering protocol suites, network stacks, and channel access procedures that could be adopted on top of the terahertz channel at the nanoscale. In this context, it appears evident that a modular and freely available simulation platform would be highly useful to support and let research activities converge towards common goals.

Currently available tools, such as NanoNS [7], N3Sim [8], and the one proposed in [9] have been explicitly conceived for diffusion-based molecular communications, thus, they are not really useful to study the promising solutions where WNSNs adopt EM waves in the terahertz band.

Motivated by this premise, herein, an innovative open source simulator for EM-based nanonetworks, namely *Nano-Sim*, is proposed. It is implemented in the Network Simulator 3 (NS-3) platform[1]. It has been designed in a modular way to allow graceful upgrades in the future by researchers interested in evaluating novel nanonetworking solutions. Moreover, it is freely available under the GPLv2 license[2].

At the present stage, the main features covered by the developed *Nano-Sim* tool are: (i) the implementation of different devices forming a WNSN; (ii) a physical interface based on the Time Spread On-Off Keying (TS-OOK) modulation scheme; (iii) a simple Media Access Control (MAC) protocol; (iv) a routing module based on the selective flooding strategy; and (v) a generic unit for generating and processing messages.

The effectiveness of *Nano-Sim* has been analyzed by studying a health-monitoring scenario, where nanodevices are diffused into an artery for collecting information about chemical particles and biological functions. Simulation results demonstrated that WNSN performance is greatly in-

[1]The NS-3 Network simulator is available at http://www.nsnam.org/.

[2]The *Nano-Sim* simulator is available at http://telematics.poliba.it/nano-sim.

fluenced by the number of nodes. In fact, an increase of the number of nanonodes leads to a higher communication reliability at the expense of delays (which remains below few nanoseconds). Finally, a scalability study on the computational requirements (i.e., simulation time and memory usage) of *Nano-Sim* has been provided, showing that, as obviously expected, the simulation time increases with the number of nanomachines, but the memory usage remains limited also in scenarios with a high traffic load.

The rest of this paper is organized as follows. Sec. II sketches open issues on WNSNs and motivates the need for *Nano-Sim*. The simulation tool is then described in Sec. III and its performances are evaluated in Sec. IV. Finally, Sec. V draws the conclusions and discusses planned upgrades of the proposed simulator.

II. Open Issues related to WNSNs

The research on WNSN is still at an embryonic phase and a number of critical issues have to be faced before a spread adoption of such a technology in biomedical, industrial, and military domains. This section summarizes open issues in this field, specifying the aid that *Nano-Sim* can contribute for solving each of them.

A. Network Architecture

A generic WNSN is composed by three kinds of nodes [10]: *nanonodes*, *nanorouters*, and *nanointerface*.

Nanonodes are tiny devices with very scarce energy, computational, and storage capabilities. They are diffused into a target area for sensing the environment. *Nanorouters* are, instead, nanodevices having sizes and resources larger than previous ones. They aggregate and process the information coming from *nanonodes* controlling their behavior by using short control messages. Finally, the *nanointerface*, which is the most complex node, inter-networks (acting as a gateway) the WNSN with the rest of the world.

All main aspects related to the communications among nanonodes, the interactions between them and nanorouters, and the interface between a WNSN and the Internet still need to be deeply studied to discover the most promising and feasible solutions [10]. To support these research activities, *Nano-Sim* models all the aforementioned kinds of nodes and provides, for each of them, basic functionalities that could be improved and customized depending on the scenario to be analyzed.

B. The Physical Layer

Recent studies in [6] demonstrate that graphene-based nanoantennas could support EM communications in the terahertz band (i.e., 0.1 $\div$ 10.0 THz), allowing bit rates extremely higher (i.e., some terabit/s) than the one obtained with other common wireless systems. On the other hand, they can guarantee only limited transmission ranges that cannot exceed few tens of millimeters.

In addition, due to size and energy constraints of nanomachines, techniques based on the transmission of signals with long duration, which are typically adopted in wireless sensor network (WSN) [11], cannot be used in a WNSN. Considering the huge available bandwidth, a promising solution could consist in exchanging very short pulses spread over the entire spectrum. That is, a sequence of logical 1's and 0's could be encoded using short duration pluses, properly spaced according to the binary word to encode. For this reason, the TS-OOK modulation scheme seems to be the most appealing technique to adopt, given that it ensures both high energy and communication efficiency [12]. With TS-OOK, a logical 1 is transmitted by using a short pulse and a logical 0 is encoded as a silence. The only limitation to its straight usage is related to the time between two consecutive pulses, which should be kept longer than the pulse duration because the communication unit can work only with a very low duty-cycle, due to technological limitations. The adoption of TS-OOK has two important advantages. Firstly, it does not require that nanodevices should be synchronized before starting the transmission of the packet. Moreover, it allows multiple users to safely share the same wireless medium; in fact, since the time between the transmission of two consecutive pulses has to be much longer than the pulse duration, several nanodevices can concurrently send sequence of pulses which are slightly time-shifted, without incurring in collisions.

In this regard, *Nano-Sim* fully support TS-OOK modulation and allows to set its main parameters (e.g., the duration, the transmission frequency, and the energy of pulses) according to the application scenario and to the upper layers of the protocol stack.

C. Channel access procedures

A WNSN is expected to be composed of a very high number of nodes. This makes the design of the channel access procedure harder than in classic wireless systems, not working at the nanoscale. In addition, complex protocols, like those requiring node synchronization, are not recommended due to the tight computational bounds nanonodes are subject to. Besides, approaches based on carries sensing strategies cannot be applied to pulse-based communications due to the lack of a signal to sense. All these peculiarities force the choice of asynchronous MAC algorithms (like the one proposed in [13]) as the best candidates for a WNSN. In fact, they allow packet transmission without any channel contention stage and, thanks to their very low computational requirements, they are also easy to implement on nanodevices.

With reference to issues related to MAC, within *Nano-Sim* we propose (for the time being) a simple asynchronous strategy that can be exploited as a starting point for the design of more complex solutions.

D. The Network Layer

Communications at the terahertz band are characterized by very limited transmission ranges (in the order of few tens of millimeters) so that establishing a multi-hop path between sender and receiver becomes a critical aspect in a WNSN.

Of course, also routing operations, as channel access and modulation procedures, should be handled with the smallest computational complexity. For this reason, the WNSN could be organized following a hierarchical architecture [10]. According to this scenario, nanonodes would be arranged in small clusters, each one having its reference nanorouter, which is in charge of forwarding nanomachines measurements to the nanointerface. It is worth to note that the creation and the management of multi-hop paths from each nanonode to its reference nanorouter is a very challenging open issue, which could be very critical in those scenarios entailing some degree of mobility.

To this end, *Nano-Sim* provides a flexible network layer that can be used also to evaluate hierarchical routing operations. In the present implementation, selective flooding is adopted (as in [3]) in each cluster of the WNSN. Due to the modular nature of *Nano-Sim*, such a solution can be easily extended with more complex routing strategies as required by the user.

III. Modeling electromagnetic-based Nano Networks

Nano-Sim has been devised as a tool for researchers working on the most relevant research topics of WNSN protocol stack design and performance analysis. It has been conceived as an open source modular tool in order to allow the community to devise, add, and test new functionalities and solutions intended for the terahertz wireless communications. The possibility of using a common reference simulator could ease the comparison of different research outcomes and avoid to waste precious human resources to be employed in the development of an entire simulator each time a novel protocol has to be proposed.

Similarly to [14], *Nano-Sim* has been written in C++ by using the object-oriented paradigm, in order to ensure modularity, flexibility, and high performance. In particular, it has been developed on top of NS-3, an emerging discrete-event and open source network simulator, designed for replacing the popular NS-2. Note that NS-3 is not an updated version of NS-2 and it is not backward-compatible with it. Its core, as well as all available modules, are written in C++. Network scenarios are no longer defined with oTcl scripts, but by using C++ or Python. A very useful feature provided by NS-3 is the possibility to interact with real systems thus supporting the integration between simulation scenarios and real test-beds. The NS-3 simulation core supports research on both IP and non-IP based networks. The large majority of its users adopt it for testing wireless systems (such as WiFi, LTE [15], and WiMAX [16]). In this work NS-3 has been considered as the best platform to be used in the implementation of *Nano-Sim* due to its high flexibility and modularity.

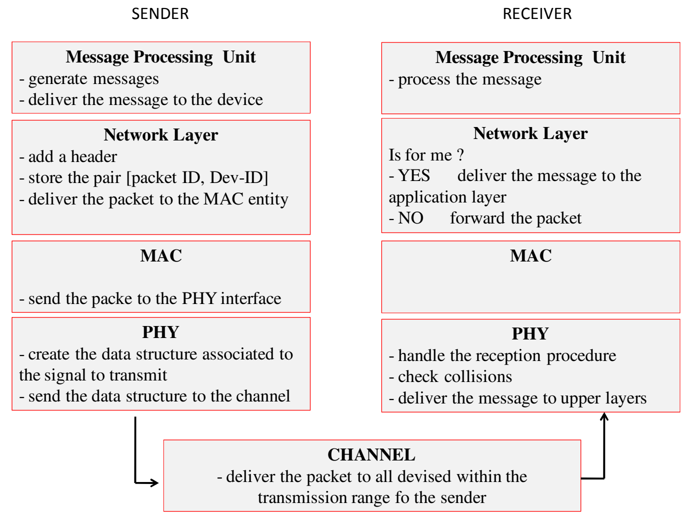

Figure 1. Network architecture entailed by *Nano-Sim*.

As reported in Sec. I, at the present stage, the main features covered by *Nano-Sim* are: (i) the implementation of devices of a WNSN; (ii) the TS-OOK modulation scheme; (iii) a simple MAC protocol; (iv) routing based on the selective flooding strategy; and (v) a generic unit for generating and processing messages.

In what follows, a comprehensive presentation of *Nano-Sim* features is reported.

A. Network entities and protocol stack

Nano-Sim supports the three different kinds of nanodevices by using a different class for each of them.

The network architecture of a generic nanomachine entailed by *Nano-Sim* is pictured in Fig. 1, where the most important procedures and their relations are shown. It is possible to observe that each device, identified by an unique *Dev-ID*, is modeled as a container of several entities, e.g., the message processing unit, the network layer, the MAC entity, and the physical (PHY) interface. With this solution, the logical model of a node reflects the real architecture of a nanomachine and its main functionalities.

Since it is difficult to adapt the TCP/IP on a WNSN, we propose a prototype protocol stack, which is general enough to be considered as a starting point for future optimized solutions. Note that the implemented protocol architecture has been conceived following all the guidelines discussed in [10].

The *Message Process Unit* generates and processes messages. A constant bitrate source is adopted to create periodically fixed-length packets (both interarrival time and packet size can be modified by the user), as it should be the case of WNSNs. Once a new message is ready to be delivered below through the stack, the *Message Process Unit* sends it to the nanodevice that, by applying sequentially routing and MAC algorithms, delivers it to the PHY interface.

The network layer handles routing operations. At the present stage, each nanomachine adopts the selective flooding strategy (as suggested in [3]) for delivering the message to the destination. A node, receiving a message from the upper layer (i.e., by the *Message Process Unit* it hosts) or from another device through the physical interface, sends the packet to all the other devices within its transmission range. In addition, we imposed that each node should not forward messages in the opposite direction of the destination.

Before delivering the packet to the MAC entity, the network layer adds a header, which is composed by four fields: the source *Dev-ID*, the destination *Dev-ID*, the packet ID, and the Time To Live (TTL). Source and destination *Dev-IDs* are set to the *Dev-ID* of the local node and of the receiver, respectively. Each of these fields is 32 bits long, as default. However, the structure of the header added by the network layer, as well as the value and the length of each field, can be modified depending on the use case. The packet ID is assigned by the *Message Process Unit* in a sequential manner, whereas the TTL is set to its default value (i.e., 1000) and it is decreased by one unit at each hop [3].

A packet, arriving at the network layer from the MAC and not directed to the receiving node, is forwarded again downstream through the stack if it has not been already forwarded in the past and its TTL is not null. Otherwise, it is discarded.

To avoid an excessive waste of bandwidth, it is very important to limit duplicate forward operations of the same packet. In this regard, we remark that each message can be uniquely identified by the pair *[packet ID, source Dev-ID]*. Hence, a node could be able to not re-transmit packets which have been already received in the past by keeping in memory the pair *[packet ID, source Dev-ID]* associated to the latest received messages. As default, each nanomachine stores information about the last 20 received packets.

At the MAC layer, a simple asynchronous strategy has been designed. It does not handle the acknowledgment and the retransmission of messages: a packet is transmitted from the network layer to the physical interface without executing any kind of control.

Furthermore, we decided to not include further headers at the MAC layer. In our opinion, in fact, all information which are useful for delivering messages to the destination node are included within the header added at the network layer.

[3]We remark that in a WNSN a message can reach the destination after traversing hundred of nanodevices; this behavior justifies a so high default TTL.

B. Channel and Physical models

Nano-Sim has been conceived as a system level simulator. The impact of the channel behavior is modeled by means of the knowledge of the physical transmission range (note that the same approach is frequently adopted by other famous simulators like NS-2 and Omnet++). During the simulation, due to the limited transmission range we expect at the terahertz band, nanomachines will be split in different broadcast domains. For this reason, the channel can deliver a packet to a given node (thus triggering the reception procedure at the PHY layer) if its distance with respect to the source does not exceed the *transmission range* threshold.

Simply modifying the open source of *Nano-Sim*, an user can exploit more sophisticated channel models, like those presented in [6] and [17], for properly setting the transmission range of nanomachines and some physical parameters. Then, focusing the attention on a specific application field of its interest, he can test performances of the WNSN and the protocol stack devised by itself from a system level point of view.

Anyway, despite the current version of the tool manages the communication among nanomachines by means of the knowledge of the physical transmission range, we designed *Nano-Sim* in order to allow future upgrades able to offer also more sophisticated channel and physical models. In this regard, classes implementing the physical interface and the channel have been inherited from those available inside the spectrum framework (i.e., another module available within NS-3), which has been developed for supporting the development of spectrum-aware physical and channel models [18].

The communication scheme we implemented is based on the TS-OOK modulation: the message is sent by means of a sequence of pulses. We remark that simulating the transmission and the reception of each single pulse would unnecessarily increase the complexity of *Nano-Sim*, due to the very high number of nodes belonging to a WNSN. For this reason, the packet transmission is handled at the system level as depicted in the following:

- the MAC entity, after the channel access procedure, calls the method for the PHY interface which is in charge of packet transmission.
- The PHY layer creates the data structure associated to the signal to transmit, storing details such as, the time instant when the transmission starts, the pulse duration, T_p, the pulse transmission interval, T_i, and the transmission duration, $txTime$. In particular, the total transmission duration, is computed as

  $$txTime = [(N-1) \cdot T_i] + T_p \quad (1)$$

  where N represents the packet length expressed in bits.
- The aforementioned data structure is delivered to the channel and then sent to all the nodes which are in the transmission range of the sender.

- The destination node handles the reception procedure. In particular, it verifies if there are physical collisions during the time interval required for receiving the packet (i.e., $txTime$). A collision occurs if pulses belonging to different transmissions overlap in the time domain. In order to detect this episode, a nanomachines stores transmission parameters (i.e., those listed at the point 2) associated to all active reception procedures. Once a reception procedure ends, the node exploits these parameters for re-building the sequence of received pulses during the time interval required for receiving the considered packet (i.e., $txTime$). In this way, it is able to evaluate the presence of overlapped pulses. If the packet is correctly received, it will be forwarded to upper layers of the protocol stack (see Fig. 1). Otherwise, all collided messages will be deleted.

C. Software architecture

Nano-Sim is built completely in C++ and its code is freely available under the GPLv2 license to boost its diffusion in the research community [19]. Fig. 2 shows the UML diagram of the most important classes that compose the module, emphasizing the relationship they have with the NS-3 core. It is important to remark that the diagram only reports the most important data members and functions. Some details about relationships among objects have been omitted due to limited space. However, with the aim of clarifying the entire software architecture, some details about classes reported in Fig. 2 can be found in In Tab. I.

IV. Performance Evaluation

Herein, we demonstrate the practical utility of the developed tool in the biomedical field, by studying a health-monitoring system based on WNSN. Then, we investigate the simulator scalability in terms of computational requirements (i.e., simulation time and memory usage) in order to understand *Nano-Sim* limits and forecasts future research.

A. Analysis of a Health-monitoring System

We suppose to have one nanodevice, which performs functionalities of both nanorouter and nanointerface, and a number of nanonodes in an artery collecting information about chemical particles and biological functions. Starting from these measurements, these nodes generate, periodically, messages sent, through multi-hop paths, to the nanorouter/nanointerface, which is connected to the health remote server by means of an IEEE 802.11 wireless connection (see Fig. 3).

In this scenario, only a quota of nanonodes (i.e., the nanosensors) are equipped with a sensing unit and are able to sense the surrounding environment, whereas remaining ones acts as simple relays. We will study the performance of this system by varying the total number of nodes and the percentage of sensors, as well.

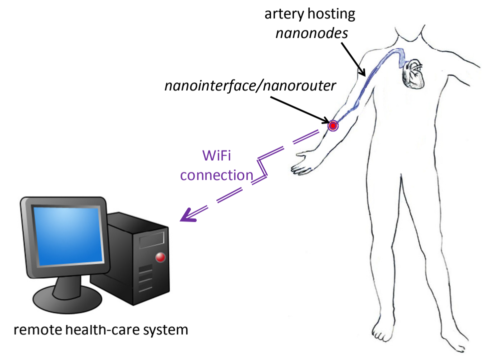

Figure 3. The simple health-monitoring system considered in simulations.

Table II
SIMULATION PARAMETERS.

Parameter	Value
System parameters	
Simulation duration	3 s
Number of Seeds	10
Number of nanonodes	from 500 to 3000
Percentage of nanosensors	10%, 50%, and 100%
Number of nanointerface/nanorouter	1
Artery size	$10^{-3} \times 10^{-3} \times 0.5\ m^3$
PHY details	
Pulse energy	100 pJ
Pulse duration	100 fs
Pulse Interarrival Time	10 ps
Transmission range	$2 \cdot 10^{-3}$ m
Network Layer	
Routing protocol	Selective flooding algorithm
Initial TTL value	1000
MessageProcessingUnit	
Packet size	128 bytes
Message generation time interval	0.1 s

Simulation settings are summarized in Tab. II. We remark that the PHY layer has been configured according to parameters suggested in [13].

Fig. 4 reports the Packet Loss Ratio (PLR) in the network, measured as the percentage of packets that are not received by the nanointerface.

The first interesting result is that the PLR decreases as the number of nanonodes increases because there are more chances to find a multi-hop path to the nanorouter. This finding is relevant and should be very carefully accounted for sizing a WNSN. In details, in scenarios with less than 2000 nanodevices, the PLR is so high that it becomes unacceptable. This result underlines that a WNSN composed by a limited number of nodes cannot guarantee the delivery of messages between sensors and the remote server, and viceversa. Moreover, we remark that the implemented MAC protocol, due to the absence of any mechanisms of message acknowledgment and retransmission, is not able to react to physical collisions, thus favoring the increment of the PLR.

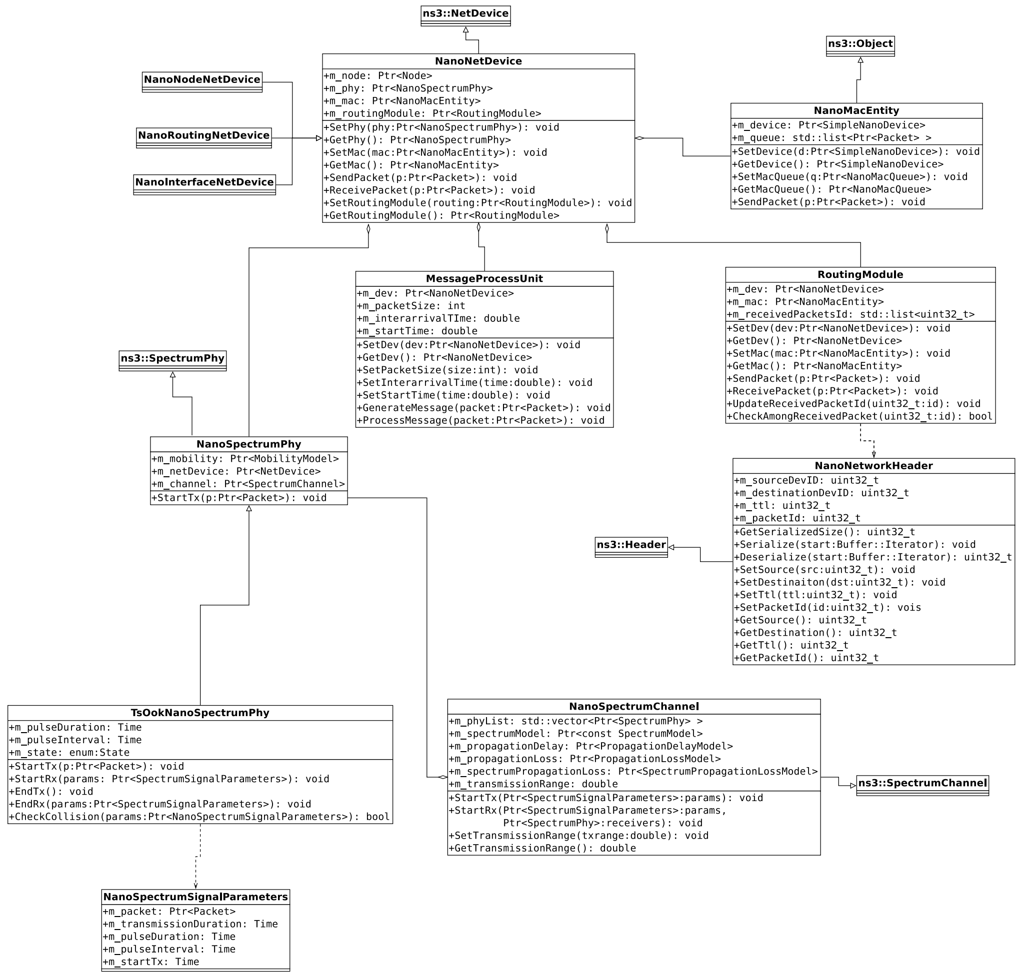

Figure 2. UML class diagram of *Nano-Sim*.

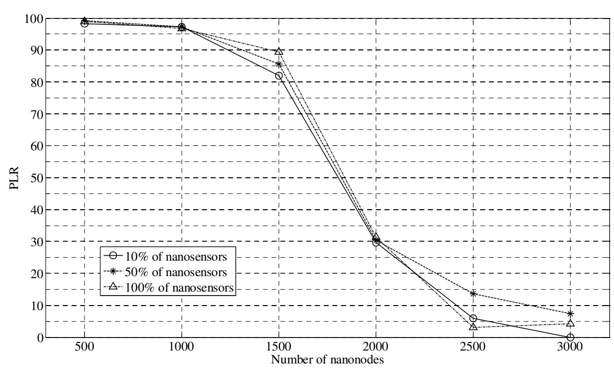

Figure 4. PLR vs number of nanonodes.

Packet delays, measured from the node that generates messages to the nanointerface, are shown in Fig. 5. They are very influenced by the number of nodes as well as by the length of routing paths established between the source and the destination. With up to 2000 nanonodes, growing the number of nodes, messages reach destination through an increasing number of hops; this produces an increment of packet delays (which are smaller than few nanoseconds). In higher load conditions, i.e., more than 2000 nanodevices, all possible path lengths are almost uniformly distributed within the network; consequently, there are no evident variations of the average delays.

Finally, we observed that the percentage of nanosensors inside the WNSN does not impact on both PLR and packet delays.

To summarize, the presented analysis demonstrates that the PLR is a critical issue in a WNSN. Therefore, sophisticated MAC and routing strategies as well as a careful tuning of the network size are required. An increase of the number of nodes leads to a smaller PLR at the expense of higher delays, so that an optimized trade-off among protocol complexity and network size is a relevant topic in WNSNs that we think *Nano-Sim* will help to solve in the future.

Table I
DESCRIPTION OF THE MAIN C++ CLASSES DEVELOPED IN *Nano-Sim*

Class	**Description**
NetDevice	Already available within the NS-3 framework, it implements all the basic functionalities of a device (the most important one is the interaction with upper and lower layers). We extended such a class for modeling all nanomachines.
NanoNetDevice	It offers features which are common to all developed nanomachines. It has been conceived as a container of all entities forming the protocol stack (i.e., the physical interface, the mac entity, and the routing module).
NanoNodeNetDevice	It provides operations for the *nanonode*.
NanoRouterNetDevice	It provides operations for the *nanorouter*.
NanoInterfaceNetDevice	It provides operations for the *nanointerface*.
MessageProcessUnit	It generates and processes messages at the application layer.
RoutingModule	It handles routing operations (i.e., the selective flooding strategy) and offers a protocol interface between the application layer and the mac entity.
NanoNetworkHeader	It is the header added by the routing module. It contains all the parameters required for delivering the message to the destination node.
NanoMacEntity	It implements a simple channel access procedure.
SpectrumPhy	It models the physical interface within the spectrum framework. We extended this for defining the physical interface of a nanomachine.
SpectrumChannel	It models the channel within the spectrum framework. We extended it class in order to create the wireless channel of a WNSN.
TsOokNanoSpectrumPhy	It represents the physical interface of a nanomachine. It handles the transmission and the reception of messages, both based on the TS-OOK modulation scheme. In addition it verifies the presence of physical collisions during the reception procedure.
NanoSpectrumChannel	It implements the channel of a WNSN.
NanoSpectrumPhysicalParameters	It is the data structure associated to the signal to transmit which details about the transmission itself are stored in (e.g., the time instant when the transmission starts, the pulse duration, the pulse transmission interval, and the transmission duration).

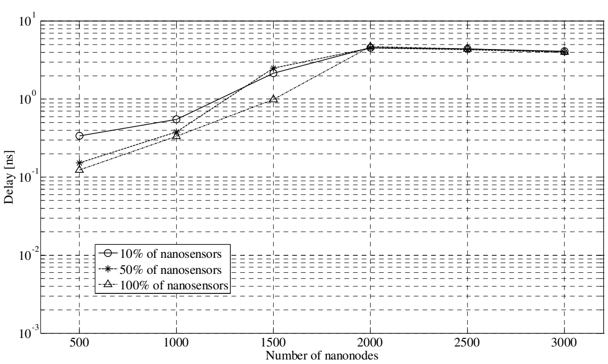

Figure 5. Average packet delay vs number of nanonodes.

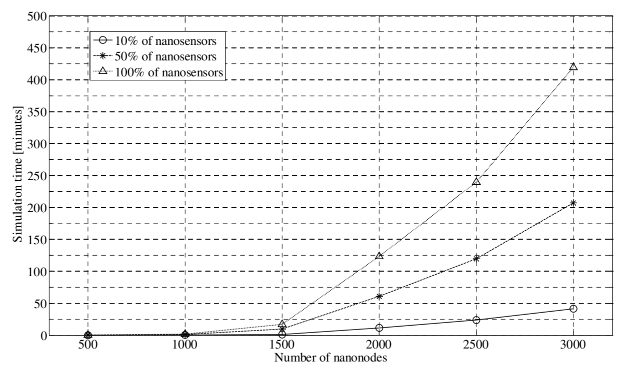

Figure 6. Simulation time vs number of nanonodes.

B. Scalability Test

To provide a further insight, we evaluate the scalability of the proposed module in terms of both simulation time and memory usage on a Linux machine with a 2.6 GHz CPU and 4 GBytes of RAM. We considered a WNSN with settings equal to those used before.

We have measured that the memory usage is always less that 250 MB and no significantly differences have been obtained when increasing the number of nodes. This is a great achievement in terms of scalability.

A different behavior has been registered for the simulation time. As depicted in Fig. 6, the higher the number of nanosensors the higher the simulation time. In particular, such a time increases exponentially. To mitigate this drawback, and considering that the highest computational effort is due to the procedure that handles the transmission of packets, we are planning to devise future upgrades that will simplify the way the simulator instance that models the channel delivers the message to PHY interfaces of nodes forming a WNSN.

V. CONCLUSION AND FUTURE WORKS

In this paper the new open source framework *Nano-Sim* has been proposed and described in order to simulate a WNSN. Its main covered features are: the implementation of different devices forming a WNSN, a physical interface based on the TS-OOK modulation, a simple MAC protocol, a routing module handling the selective flooding strategy, and a generic unit for generating and processing messages. Its effectiveness has been evaluated by analyzing a health-monitoring system and by investigating its computational requirements. The network architecture of nanodevices, as well as the entire protocol suite, have been designed in a modular way, to enable graceful future upgrades. An user, in fact, could be able to add more sophisticated network operations and WNSN architectures, modify existing routing and channel access procedures, or design novel approach as well, introduce headers for storing further parameters and supporting more effective algorithms in any points of the protocol stack, conceive different communication strategies at the physical layer, and so on. We believe that *Nano-Sim* has all the characteristics to become a reference tool for researchers working in the area of nanonetworks. As next steps of our work, we plan to extend the simulator by implementing new features, e.g., more sophisticated routing, MAC, and PHY protocols and models, as well as simplified procedures for handling packet transmissions. Finally, more complex applications in both medical and industrial fields will be also investigated.

ACKNOWLEDGMENT

This work was supported by ERMES PON 01_031133 and DSS PON 01_024993 projects, both founded by the Italian MIUR.

REFERENCES

[1] I. Akyildiz and J. Jornet, "Electromagnetic wireless nanosensor networks," *Nano Communication Networks*, vol. 1, no. 1, pp. 3–19, Mar. 2010.

[2] J. Jornet and I. Akyildiz, "The internet of multimedia nano-things," *Nano Communication Networks*, vol. 3, no. 4, pp. 242 – 251, Dec. 2012.

[3] B. Atakan and O. B. Akan, "Carbon nanotube-based nanoscale ad hoc networks," *IEEE Communications Magazine*, vol. 48, no. 6, pp. 129 –135, Jun. 2010.

[4] S. Abadal, E. Alarcn, M. C. Lemme, M. Nemirovsky, and A. Cabellos-Aparicio, "Graphene-enabled wireless communication for massive multicore architectures," *IEEE Commun. Mag.*, 2012, to be published.

[5] I. F. Akyildiz, J. MiquelJornet, and M. Pierobon, "Propagation models for nanocommunication networks," in *Proc. of the Fourth European Antennas and Propagation Conference, EuCAP*, Apr. 2010, pp. 1 –5.

[6] J. Jornet and I. Akyildiz, "Channel modeling and capacity analysis for electromagnetic wireless nanonetworks in the terahertz band," *IEEE Transactions on Wireless Communications*, vol. 10, no. 10, pp. 3211 –3221, Oct. 2011.

[7] E. Gul, B. Atakan, and O. Akan, "Nanons: a nanoscale network simulator framework for molecular communications," *Nano Communication Networks*, vol. 1, no. 2, p. 138156, Oct. 2011.

[8] I. Llatser, I. Pascual, N. Garralda, A. Cabellos-Aparicio, and E. Alarcon, "N3sim: a simulation framework for diffusion-based molecular communication," *IEEE Technical Committee on Simulation*, vol. 8, pp. 3–4, 2011.

[9] L. Felicetti, M. Femminella, and G. Reali, "A simulation tool for nanoscale biological networks," *Nano Communication Networks*, Oct. 2011.

[10] I. Akyildiz and J. Jornet, "The internet of nano-things," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 58 –63, Dec. 2010.

[11] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized protocol stack for the internet of (important) things," *IEEE Commun. Survs. Tut.*, 2013, to be published.

[12] J. Jornet and I. Akyildiz, "Information capacity of pulse-based wireless nanosensor networks," in *Proc. of IEEE Conf. on Sensor, Mesh and Ad Hoc Communications and Networks, SECON*, Jun. 2011, pp. 80 –88.

[13] J. M. Jornet, J. C. Pujol, and J. S. Pareta, "PHLAME: A Physical Layer Aware MAC protocol for Electromagnetic nanonetworks in the Terahertz Band," *Nano Communication Networks*, vol. 3, no. 1, pp. 74 – 81, 2012.

[14] G. Piro, L. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE Cellular Systems: An Open-Source Framework," *IEEE Trans. Veh. Technol.*, vol. 60, no. 2, pp. 498 –513, Feb. 2011.

[15] G. Piro, N. Baldo, and M. Miozzo, "An LTE module for the ns-3 network simulator," in *Proc. of Workshop on NS-3 (held in conjunction with SIMUTools 2011)*, Barcelona, Spain, Mar. 2011.

[16] M. A. Ismail, G. Piro, L. A. Grieco, and T. Turletti, "An improved IEEE 802.16 WiMAX module for the ns-3 simulator," in *Proc. of SIMUTools 2010*, Mar. 2010.

[17] J. Jornet and I. Akyildiz, "Channel Capacity of Electromagnetic Nanonetworks in the Terahertz Band," in *Proc. of IEEE Int. Conf. on Commun., ICC*, May 2010, pp. 1 –6.

[18] N. Baldo and M. Miozzo, "Spectrum-aware channel and phy layer modeling for ns3," in *Proc. of ICST VALUETOOLS*, Pisa, Italy, October 2009.

[19] G. Piro, "*Nano-sim* - The open source framework for simulating EM-based WNSNs," [OnLine] Available: http://telematics.poliba.it/nano-sim.