Parameter study and characterization of wireless nanonetworks through simulation

(Invited Paper)

Nicolas BOILLOT, Julien BOURGEOIS and Dominique DHOUTAUT
Institut FEMTO-ST (UMR 6174) / Université de Franche-Comté (UFC) / Centre National de Recherche Scientifique (CNRS)
1 Cours Leprince-Ringuet - 25200 Montbéliard, FRANCE
Email : {nicolas.boillot, julien.bourgeois, dominique.dhoutaut}@femto-st.fr

***Abstract*—Nanowireless electro-magnetic communication networks in the Terahertz band have raised interest in the networking community these very last years. However, if detailed studies have been published on analytical modelling of these networks, no simulation have been run to study in detail the characteristics of the transmission medium. We have designed Vouivre, a standalone nanowireless simulator, which is interfaced with two micro-robots simulator DPRSim and VisibleSim. This paper describes briefly Vouivre and presents first metrics of the communication channel using a communication paradigm called TS-OOK (Time Spread On-Off Keying).**

I. Introduction

In the global miniaturization trend, electro-magnetic (EM) wireless nanonetworks are the next frontier to reach for scaling down communication networks. There have been analytical work proposed for defining and characterizing this new type of communication channel [1] which has led to the definition of a pulse-based communication scheme named Time-Spread OnOff Keying (TS-OOK) [2]. We have proposed a simulator called Vouivre[3], [4] which is able to simulate an EM wireless nanonetwork. Vouivre is plugged with DPRSim [5], a micro-robot simulator and will soon be plugged with VisibleSim [6], the successor of DPRSim. In this article, we propose to use Vouivre in his new standalone mode to study some parameters of the nanonetwork channel. More particularly, we studied different scenarios of ratio between the time of pulse and the time between two pulses called β in [2].

II. Context

A. Specificities of electromagnetic nanonetworks

Nano-wireless communications make use of nano-meter scale antennas, and many aspects of communications are consequently affected. At that scale, classical modulation schemes are difficult to implement and pulse-based communications are preferred. Time-Spread OnOff Keying (TS-OOK) [2] uses a simple coding where the presence of a pulse carries a ”1” value and its absence carries a ”0”. Extremely short pulses (~100 femtoseconds long) spread their small energy - in the order of a picojoule - into an extremely wide channel, up to 10 THz large.

The total throughput is thus potentially extremely high, in the order of terabits/s. For a given communication, each symbol is separated from the next by an interval orders of magnitude longer than a pulse. If Tp is the duration of a pulse and Ts is the time between symbols, β is the ratio Ts/Tp and a value of 1000 would not be uncommon. Because of this ratio, the achievable throughput for a given transmission is also orders of magnitude lower than the total theoretical throughput.

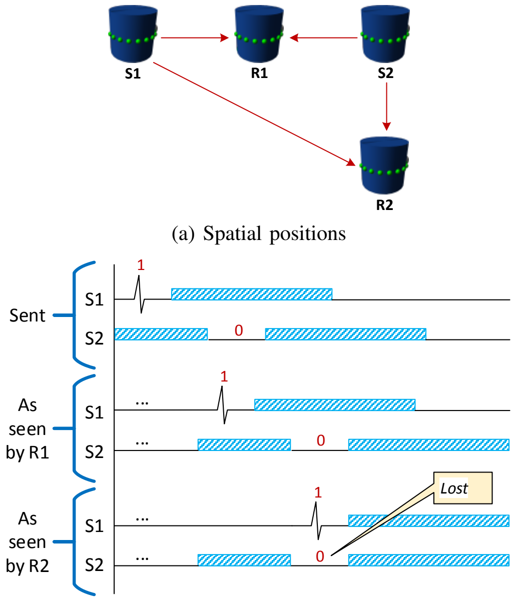
But a very interesting property of such a coding scheme is the ability to multiplex transmissions over time. Transmitters can indeed send their symbols during the inter-symbol time of the other transmitters. Collisions may occur, but their probability is directly dependent to the value of β and the number of concurrent transmissions. The multiplexing capability increases with the value of β.
Of course, higher level coding schemes are required to detect and eventually correct errors. The required redundancy will accordingly reduce the effective data throughput, but because it was so high to begin width, its value can still be kept quite high in most cases.

Such a wide channel behave peculiarly compared to more common wireless technologies. In particular various molecules in the environment will absorb parts of the spectrum while keeping others relatively untouched [7]. The energy absorbed by the environment is radiated again and essentially acts as noise on the channel. This phenomenon is known as equivocation. The noise on the channel rapidly increases with distance and transmission power, and rapidly surpass the signal power at the receiver. This is happening even when no parallel transmission are occurring and limits the effective communication range. Communications ranging up to tens of centimeters seems possible

Because the pulses are so short (~100 femtoseconds), the propagation delay (~3 nanoseconds per meter) can be greater than the duration of a pulse and as such collisions can easily occur at receiver side even if symbols were sent at different times. Figure 1 shows how symbols can overlap from a receiver point of view even if they were not sent at the same time. Sender S1 sends a ”1” (as a pulse) and S2 later send a ”0” as an absence of pulse. Because receiver R1 is at the same distance from S1 and S2, it received those symbols one after the other and decodes them correctly. Receiver R2 is much closer from S2 than from S1. As the propagation delay from S1 is longer, the two symbols overlap and the ”0” from S2 is masked by the ”1” from S1. Also, if the nodes are able to move, the propagation delay will not remain constant over time for a given communication link.

However, it is interesting to note that only in certain circumstances a symbol may be corrupted by a collision, thus alleviating a little this problem. Only when receiving a ”0” can a node be affected by a collision. If a node is currently receiving a ”1”, whatever it receives at that time from others nodes, it will still be interpreted as a ”1”. When receiving a ”0”, receiving other ”0” will not cause a problem either. Only concurrents ”1” will mask a ”0” currently being received, as it would be the case for receiver R2 for the symbol sent by S2 on Figure 1.

This work has been funded by the Labex ACTION program (contract ANR-11-LABX-01-01) and ANR/RGC (contracts ANR-12-IS02-0004-01 and 3-ZG1F) and ANR (contract ANR-2011-BS03-005)



(a) Spatial positions

(b) Chronogram

Fig. 1. ”0” symbols can be masked by ”1” even if not sent at the same time

B. *Claytronics and wireless nanonetworks*

This work takes place in the Claytronics project where MEMS micro-robots called catoms are used to create programmable matter. Catoms are covered with structures called ”features”. Those features are used as mean of both attachment (by electromagnetic or electrostatic force) and direct communication. By actuating their features, catoms are able to move around their direct neighbors similarly as a stepper motor but there are also able to communicate. Wireless communications network would enhance the way catoms are communicating [3], improving the communication range and allowing easy way of broadcasting information.

A Claytronics ensemble is a dense environment with many catoms in the communication range and most control algorithms benefits from broadcasting capability, speeding information diffusion process, allowing neighbor discovery and providing ability to communicate with non contiguous groups of catoms. Broadcasted messages can also be used as a form of synchronization, which is very useful when coordinating movements of groups of catoms.

In Claytronics environment, energy is provided to each catom from an external power source, and if required can be routed from catom to catom. This is different from sensors networks where their communications are much more constrained due to the rate they gain energy. In sensors network, nodes usually rely on battery or have to harvest energy from their environment.

III. Vouivre, the simulation environment

Vouivre [3] is a C++ network simulation library we developed as both an extension for DPRSim and as a standalone simulator. Whereas DPRSim makes use of a time-slicing approach (time is advancing step by step, and each duration of a step is constant), Vouivre is based on the discrete-event paradigm more commonly encountered in other network simulators like NS3 [8], OMNeT++ [9] or OPNET [10] to cite a few. Discrete-event simulators do not offer physics and real-world modeling like we are doing in conjunction with DPRSim: for example, mobile network nodes are only 2D objects with no support for collision.

One of the major problem when simulating nanonetworks is to provide fast simulation. As we have seen a pulse duration in expressed in femtoseconds, so simulating few seconds of communications will require a huge number of events. To speedup the simulation, we only compute changes in the number of concurrent transmissions affecting the reception of an incoming packet. Each change is named a pivot. Each couple of pivots determines a periods in which each incoming packet is affected by a coefficient. This coefficient is retrieve from a precomputed table generated from the statistical model of the network capacity done by Jornet and Akyildiz. This model takes into account the number of concurrent transmissions along with the distance from the sender. The path-loss and noise in the Terahertz Band are computed by using models introduced in [7], [1]. A standard medium with 10% of water vapor is considered. This model takes into account the number of concurrent transmissions along with the distance from the sender. We used an equal probability of ”0” and ”1” but we will see in the experiment that for high (thousands and above) numbers of communicating nodes a low-weight adaptative coding would enhance the network capacity.

Figure 2 shows the point of view - affected by the propagation delay - of a node simultaneously receiving three messages. The impact of concurrent transmissions on the various parts of the messages has to be computed as the average number of bits correctly received. During the first period (period A), only ”message 1” is being received and got a factor of 1 from the channel capacity table. This table takes into account the distance and the equivocation effects of the message itself on the channel. Getting a factor of 1 is possible, but even alone on the channel, depending on the distance it could be less. The second period (period B) sees two concurrent transmissions, which get different coefficients because the distance to the transmitter is not the same. The third period (period C) sees more transmitters and consequently even smaller factors. When the reception of a packet ends, a global factor for this packet is computed proportionally from the factors of its individual periods. For each period (computed from a couple of consecutive pivots), the information rate in bit per symbol is extracted from the statistical MATLAB model of the network capacity done by Jornet and Akyildiz (cf [1] and [4]). This bit information rate is then used to compute the SCORE of the message receiving which is the average of the information rate weighted by period lengths relatively to the whole message length.

On such an unfavorable environment, the probability for a symbol to be lost or corrupted is usually great. Coding schemes adding resilience are necessary, but can’t get too complex because of limited processing power on nano-devices. To determine whether a message with corrupted symbols can be corrected or not, a Majority Vote Takers (MVT) algorithm [11], [12] has been implemented. The higher the redundancy factor is, the bigger the message size on the transmission medium will be (see Figure 3). As a consequence, it is necessary to take into account this phenomenon which shifts pivots for the score calculation of incoming messages.

Figure 3 shows the point of view of a receiver (after taking into account the propagation delay) receiving simultaneously 3 messages without redundancy (upper part of the figure). The lower part of the figure highlights the same situation with a redundancy factor of 3.

Depending on the redundancy factor, pivots used to calculate scores for incoming messages are modified. As a consequence, the different messages are overlapping in a different way modifying the score calculation. Furthermore, depending on the redundancy algorithm used and on the position of the redundancy circuits, the ability to correct message error can also be affected. Considering these two

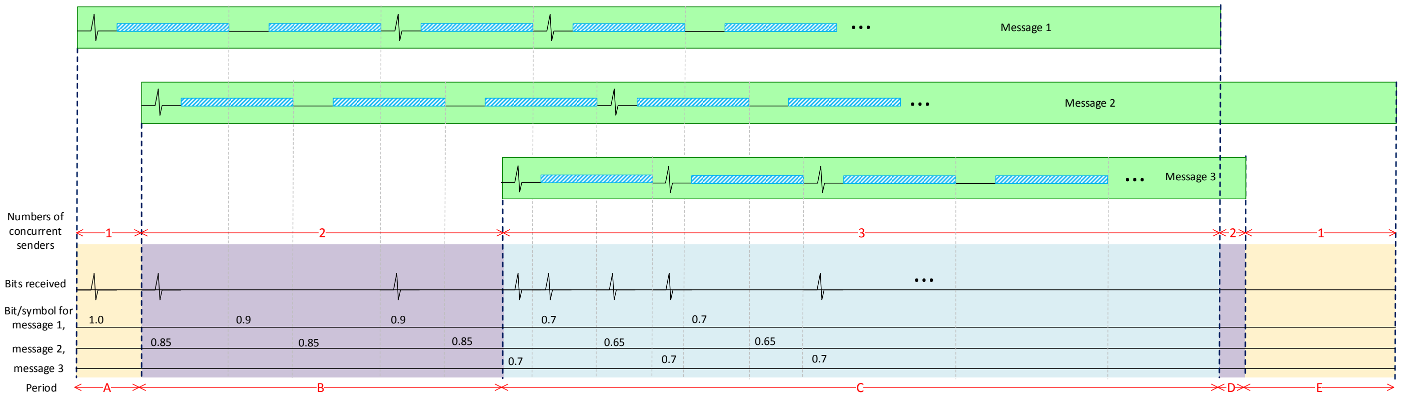


Fig. 2. Calculating the number of bit received per message by using the information rate.

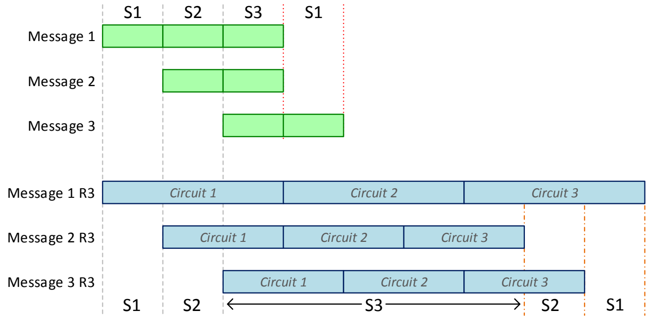


Fig. 3. Modifying the scoring system to support N-modular redundancy: Redundancy circuits are shifting pivots which requires a new scoring calculation

points, it is not possible to calculate the final score of the messages using the usual formula for N-modular redundancy described in [11], [12]. Indeed, these formula are using an equiprobability for all redundancy circuits which is not the case in our situation. For each redundancy factor, it is necessary to determine the probability of success of the majority vote from all possible vote combinations. For instance, with a redundancy factor of 3 and considering P_{c1}, P_{c2}, P_{c3} as individual success vote probabilities for the 1st, 2nd and last redundancy circuit, the probability of success of the vote of the majority circuits is given by the following formula :

$$\begin{aligned} P_{r3} =& (1 - P_{c1}) * P_{c2} * P_{c3} \\ &+ P_{c1} * (1 - P_{c2}) * P_{c3} \\ &+ P_{c1} * P_{c2} * (1 - P_{c3}) \\ &+ P_{c1} * P_{c2} * P_{c3} \end{aligned}$$

We have generated the probability of success using the same way for all the modular redundancies we have used.

IV. Simulation Results

The scenario detailed here aims to study the average of aggregated useful throughput achievable by the nano-wireless communications simulation. Catoms are distributed in two distinct groups separated by a distance of 1 mm. The first one is the group of senders. The second group is reduced to one catom which plays the role of a receiver (see Figure 4). Sender catoms are separated by a distance of 1 pm which is intended to concentrate all catoms of the group in the same spatial location. As a consequence, all senders are located at the same distance from the receiver that allows to deduct that packet losses are due to collisions. Sender catoms continuously send messages with a constant useful size (28 bytes). Each sender is able to send only one message at a time. As explained in the previous section, the average number of received bits per message, for a given receiver, is calculated using a SCORE given to each concurrent incoming message (See Figure 2). To avoid threshold effects and to statistically smooth the results for all the simulations, a random message acceptance decision function is used. Thus, the score for an incoming message is processed using a probability. Messages having high scores have better chances to be received whereas messages having low scores are likely to be lost. The information rate in bits per symbol will change as explained in [1] depending on the number of nodes contained in the group of sender catoms. We configured this MATLAB model for an environment compounded of air with 10% of moisture. Three information rate matrices have been exported with different inter-symbol durations represented by the β-parameter with values 500, 1000 and 2500 (see Figure 5).

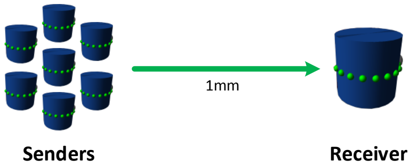


Fig. 4. Nodes spatial position for the test scenario

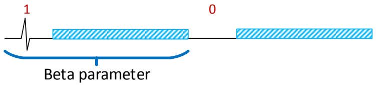


Fig. 5. The β parameter in the TS-OOK model

In Vouivre, simulation duration is set to 200ns for all simulations. Note that depending on β, each sender catom can send more or less messages during the simulation. The number of concurrent senders is also defined at the beginning of each simulation and does not evolve during simulation. When the simulation ends, the number of received messages is computed on each node to determine the average useful throughput reached during this simulation. After a sender catom loses 3 messages, it increases, at application level, the redundancy factor using to encode messages. First messages are sent without redundancy but redundancy can be increased up to a factor 11 times. We conducted several hundred independent simulations with different number of senders for three values of β exported from the MATLAB matrix.

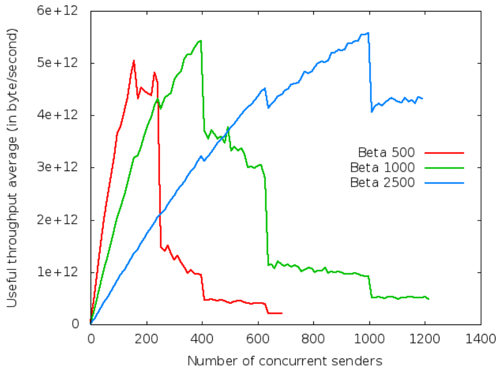


Fig. 6. Average aggregated useful throughput after 200ns of simulation

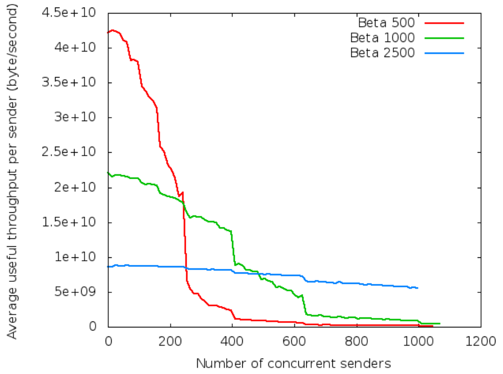


Fig. 7. Useful throughput per user related to number of concurrent senders

For each studied value of β, Figure 6 shows the respective useful throughput according to the number of senders. We can see that for a β value set to 500 and for 400 concurrent senders, more than 80% of the messages are lost, despite the redundancy factor increasing. The Figure 8 shows the lost ratio for packets related to the sent packets which is finally the packet loss rate. The higher the value of β is, the better the loss rate will be because fewer collisions occur. However, it is not desirable to use a high β (2500) for a small number of senders because the maximum throughput will not be reachable as we can see in 6 and in 7. These results suggest that it is necessary to dynamically adapt the inter-symbol duration in the TS-OOK model to follow the number of concurrent senders. Useful individual user throughput (cf Figure 7) confirms the same thresholds as in the Figure 6.

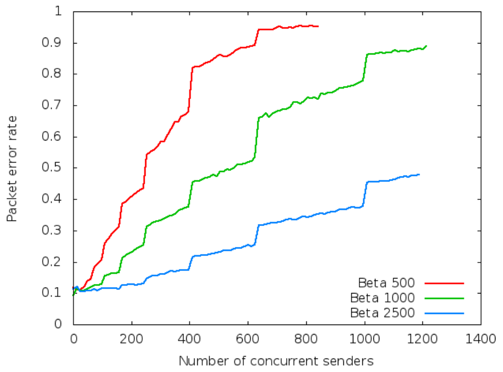


Fig. 8. Packet error rate after 200ns of simulation

Figure 9 shows the evolution of the average redundancy factor during the simulation in function of the number of concurrent senders. With small values of β (500 for instance), the redundancy factor increases when few catoms are concurrently sending. It is also interesting to note that the redundancy factor increases more slowly with high values of β because fewer messages are sent during the same simulation duration. As a consequence, the augmentation of the redundancy factor (when 3 messages are lost) is delayed. With a β of 2500, this behavior does not appear as the channel capacity is underexploited regarding the number of users.

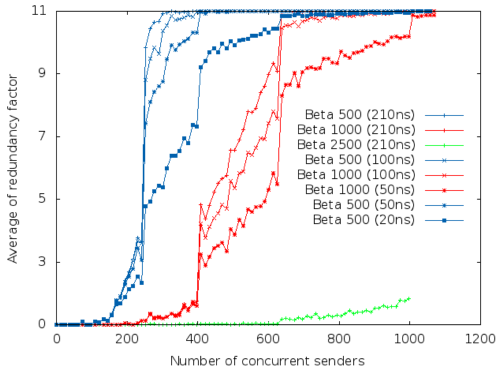


Fig. 9. Redundancy factor related to number of concurrent senders at different simulation dates

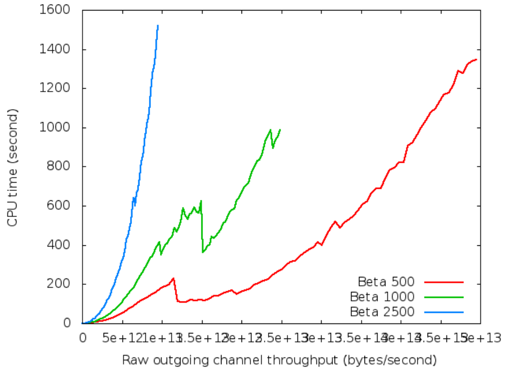


Fig. 10. Simulator efficiency related to simulation parameters

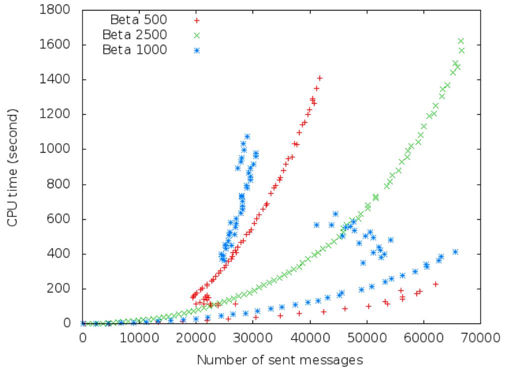


Fig. 11. CPU time required related to the number of received messages

Several distinct phenomena affect the useful throughput during the simulations:

- increasing the redundancy factor during simulations. The longer the simulations are, the more the useful throughput per user tends to 0.
- the beta value intrinsically limits the reachable raw throughput per user.
- the number of concurrent senders. Since the distance between the senders and the receiver is fixed, the SCORE of incoming messages is strongly related to the information rate.

As a consequence, the computation time is highly related to the evolution of these parameters during simulations. Since the simulation time has been set to 210ms, the number of raw sent messages is different for each simulation. The algorithm implemented to calculate the SCORE of incoming messages is interesting when periods between 2 consecutive pivots are large (several bits). But, performance problem can arise when lots of small messages are simultaneously received because in this case, the periods between 2 consecutive pivots are reduced below the length of a bit. With 1000 concurrent sender and a fixed size message of 224 bits (28 bytes), there will be about 2 x 1000 pivots per incoming message. Then, only 0,112 bit per period will be computed. In these circumstances, it should be more advantageous, in terms of performance, to use another algorithm to directly simulate pulses and thus increase simulation accuracy. By also considering the number of simultaneous receivers, it reveals a $\mathcal{O}^3$ complexity.

We must also understand that the complexity of formulas used to calculate circuit redundancy probabilities increases exponentially depending on the redundancy factor used. For instance, the formula of the 5-modular-redundancy contains 16 additions; the formula of 7-modular-redundancy contains 64 additions and the formula of 11-modular-redundancy contains 1024 additions. Furthermore, the number of factors composing each additions also increases with the redundancy factor. Redundancy increasing has a high cost on simulation performance.

Simulations have been run on a 3.2Ghz CPU core (i7-3930k). The figure 10 shows the computation time required in function of the raw channel throughput. The raw throughput sent to channel depends on the number of concurrent senders and on the β parameter, sets at the beginning of simulation. It is therefore an image reflecting the scalability of the simulator. The figure 11 shows the computation time required in function of the number of messages received. With the β sets to 500 and 1000, we can see bijections whereas with β sets to 2500, we cannot. These bijections show the redundancy cost because increasing the redundancy implies using more complex formulas and also reducing the number of sent messages (and so the number of received messages).

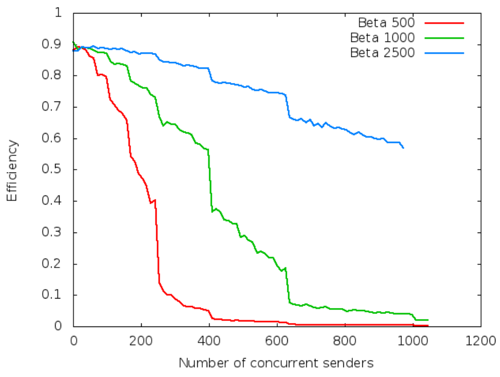


Fig. 12. Ratio between useful throughput and raw channel capacity

Figure 12 shows the ratio between the useful throughput and the raw channel capacity related to the number of users. This figure have some similarities with the Figure 8 but takes into account the evolution of the redundancy during the simulation. The useful throughput decreases when the simulation is progressing, as the redundancy factor increases.

V. Conclusion

In this article, we have described Vouivre a discrete-event wireless network simulator which comprises a module for nanonetworks simulation. The model used for the simulations is derived from the work of Jornet and Akiyldiz on TS-OOK and channel capacity of wireless nanonetworks. To speed up the simulation we used a mix between pre-calculation of parameters and simulations. We have reported the first simulations on nanonetworks showing two things. First, the parameters of the physical layer have a great influence on the performance of the nanonetwork. Decreasing the interval between pulses, called β, greatly enhance the available throughput per node while increasing β, greatly enhance resistance to errors. Second, putting in place a N-redundancy voter module at the application layer without modifying information code is quite inefficient. Once again, this has to be done at the physical layer taking advantage of a low-weight coding. As future works, we will study more parameters of wireless nanonetworks that could enhance their usage. Meanwhile, we will design a more precise nanonetworks simulation module to be able to run simulation at the pulse level.

VI. Acknowledgments

We would like to thank Dr. Josep Miquel Jornet and Prof. Ian Akyildiz for sharing with us their models of the Terahertz band.

References

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

[2] J. M. Jornet and I. F. Akyildiz, “Low-weight channel coding for interference mitigation in electromagnetic nanonetworks in the terahertz band,” in *Proc. of IEEE International Conference on Communications (ICC)*, 2011.

[3] N. Boillot, D. Dhoutaut, and J. Bourgeois, “Efficient simulation environment of wireless radio communications in mems modular robots,” in *iThings 2013, IEEE Int. Conf. on Internet of Things*, Beijing, China, Aug. 2013, pp. 638–645.

[4] ——, “Using nano-wireless communications in micro-robots applications,” in *NANOCOM 2014, 1st ACM Int. Conf. on Nanoscale Computing and Communication*. Atlanta, Georgia, USA: ACM, May 2014, pp. 1–9.

[5] B. D. Rister, J. Campbell, P. Pillai, and T. C. Mowry, “Integrated debugging of large modular robot ensembles,” in *ICRA*, 2007, pp. 2227–2234.

[6] D. Dhoutaut, B. Piranda, and J. Bourgeois, “Efficient simulation of distributed sensing and control environments,” in *iThings 2013, IEEE Int. Conf. on Internet of Things*, Beijing, China, Aug. 2013, pp. 452–459.

[7] J. Jornet and I. Akyildiz, “Channel capacity of electromagnetic nanonetworks in the terahertz band,” in *Communications (ICC), 2010 IEEE International Conference on*, 2010, pp. 1–6.

[8] T. Henderson, S. Roy, S. Floyd, and G. Riley, “ns-3 project goals,” in *Proceeding from the 2006 workshop on ns-2: the IP network simulator*. ACM, 2006, p. 13.

[9] A. Varga and R. Hornig, “An overview of the omnet++ simulation environment,” in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, ser. Simutools ’08. ICST, 2008, pp. 60:1–60:10.

[10] “http://www.opnet.com/products/modeler/.”

[11] J. Knox-Seith, “A redundancy technique for improving the reliability of digital systems,” *Stanford Electronics Laboratory*, 1963.

[12] R. E. Lyons and W. Vanderkul, “The use of triple-modular redundancy to improve computer reliability,” *IBM Journal*, pp. 200–209, 1962.