SELF-CONFIGURING ASYNCHRONOUS SLEEPING IN HETEROGENEOUS NETWORKS

Ali Medlej[1], Eugen Dedu[1], Kamal Beydoun[2], Dominique Dhoutaut[1]
[1]FEMTO-ST Institute/CNRS, Univ. Bourgogne Franche-Comté, Montbéliard, France, [2]L'ARICoD Laboratory, Faculty of Sciences I, Lebanese University Beirut, Lebanon

NOTE: Corresponding author: Ali Medlej, ali_ghasswan.medlej@univ-fcomte.fr

Abstract – *Nowadays, the heterogeneous wireless nano-network topology becomes a need for diverse applications based on heterogeneous networks composed of regions of different node densities. In Wireless Nano-networks (WNNs), nodes are of nano-metric size and can be potentially dense in terms of neighbouring nodes. Nano-nodes have limited resources in terms of processing, energy and memory capabilities. In nano-network(s), even in a communication range limited to tens of centimeters, thousands of neighbours can be found. We proposed a fine-grained duty-cycling method (sleeping mechanism), appropriate to nanonodes, which aims to reduce the number of receptions seen by a node during data packet routing. The present study reveals the usefulness of implementing the sleeping mechanism in heterogeneous networks, as well as configuring a dynamic awaken duration for nodes based on a density estimation algorithm. We also proposed an algorithm that helps in increasing the reliability of the packet received by the destination node.*

Keywords – Density estimation, heterogeneous nano-network, sleeping mechanism, routing

1. INTRODUCTION

Nano-network(s) are made up of tiny nodes, called nanonodes, of a nanometric size. These nodes use electromagnetic waves from the THz band (0.1–10 THz) [1] for their communication. Nano-nodes possess sensors, actuators, a processor, and a memory. They can move and communicate with each other [2]. These nodes are limited in their computing capabilities, storage, and energy. Therefore, they need to collaborate with other nodes to fulfil their tasks [3].

1.1 Modulation and routing techniques

Given the very limited energy of nanonodes, a modulation technique based on femtosecond pulses for terahertz communications called Time Spread On-Off Keying (TS- OOK) was proposed by Jornet and Akyildiz [4]. TS-OOK is based on femtosecond-long pulses where communica- tion data is sent using a sequence of pulses interleaved by a randomly selected constant duration (Fig. 1). TS-OOK uses an electromagnetic pulse of duration T_p to transmit a bit “1”, and silence (no transmission) for bit “0”. The time between two consecutive bits is T_s. Due to hardware and power constraints, the spreading ratio $\beta = T_s/T_p$ can be very large. Using T_p = 100 femtosecond, the total available bandwidth in the network is very high, in the order of terabit per second.

In very dense nano-network(s), a nanonode cannot frequently update its neighbourhood list. Also, the communication range is short and the desired destination node is often beyond its communication range. Therefore, it is necessary to have a routing protocol that performs the tasks of data routing between nodes. A traditional addressing scheme like IP is useless. An alternative addressing way must be used. For that, a spatial addressing routing protocol that forwards messages *in the direction* of a destination could be used.

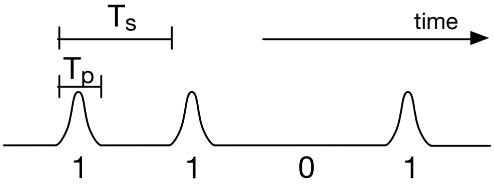


Fig. 1 – TS-OOK modulation.

Stateless Linear-path Routing (SLR) [6] is a spatial addressing and routing protocol we use in our evaluation. It implements a coordinate-based routing, in which data packets are routed in a linear routing path. The SLR protocol has two phases: (1) initial/addressing phase and (2) routing phase.

1.2 Applications and node limitations

Nano-network(s) can be potentially used in a wide range of applications and fields such as: (1) health field (e.g. monitor human vital signs, drug delivery systems), (2) military field (e.g. border control to prevent any penetration), (3) agricultural field (e.g. monitoring the temperature, humidity, and water level of plants), (4) telecommunication sector [5]. In such applications the network topology is often heterogeneous. A heterogeneous nano-network is a network composed of zones of different densities. Nowadays, many applications use this type of networks. For example, in agricultural applications, some areas have high nodes' density while others have a small density due to the varied geographical terrains. Also, the human body is considered as a natural heterogeneous network because of its internal and external structure.

An even more severe limitation (and a main difference from traditional wireless networks) lies in the inability of a single node to process the whole channel capacity. In the typical networks, a node may always listen to the channel and take whatever packet concerns it. There is no way for a tiny, energy constrained nanonode to sustain Tb/s traffic. This translates into a much smaller effective channel capacity, as all nodes in an area may saturate their reception capability way before saturating the channel itself. A typical solution to this problem is to let the nodes sleep from time to time. A traditional sleeping mechanism cannot be applied in a nano-scale network. Our proposition differs significantly from the previous schemes that make nodes sleep or awaken for periods longer than a packet. To cope with the very specific characteristics of nanonet- works, our idea is to make the node inactive for a fraction of T_s (the time between two consecutive bits). We then make it awake in order to receive the intended bit. Such a mechanism aims to preserve node resources (CPU, mem- ory, energy), therefore extending network lifetime.

In this paper: (1) we show that the sleeping mechanism works in a heterogeneous nano-scale network, (2) we propose a method to automatically tune the awaken duration of a node depending on the average density of neighbours (awakenNodes), (3) we propose an algorithm for packet retransmission at the destination zone, and (4) we test this method and show its impact on increasing received packets by the destination node.

The rest of this paper is organized as follows. Section 2 introduces applications used within heterogeneous networks. Section 3 describes the algorithms used in the paper. Packet retransmission algorithm is explained in the Section 4. Evaluation via simulations takes place in Section 5. Finally, we conclude in Section 6.

2. APPLICATION FIELDS

Different approaches in the literature focus on designing and developing routing protocols for heterogeneous electromagnetic nanonetworks. Nowadays, the nano-scale communication technology enters into many important sectors such as health, military, and agriculture.

2.1 Health sector

Heterogeneous networks pervade many applications where nodes are distributed depending on the environment conditions. Due to their nanoscale size, nanonodes can be deployed at different scales [7] inside the human body and across a variety of environmental contexts. Nanonode uses are not limited to sensing and monitoring human vital signs, but it may cover some other operations when needed.

An example of a medical application that can be assigned to nanonodes and needs high reliability and accuracy is *drug delivery* (Fig. 2). Heterogeneity resides in the human body through the variation in the size and shape of the human organs [8]. Nanomedicine may be defined as the use of nano-devices and nanostructures for monitoring, repairing or controlling human biological systems at the molecular level [9]. The nanonodes spreading in blood vessels can monitor the glucose level, and at the same time release the insulin to regulate the glucose level. The risk lies when the patient forgets to take his medicine especially those related to the control of certain levels of enzymes in the blood (as in the example above). With this type of technology, the patient becomes more able to con- trol his health troubles.

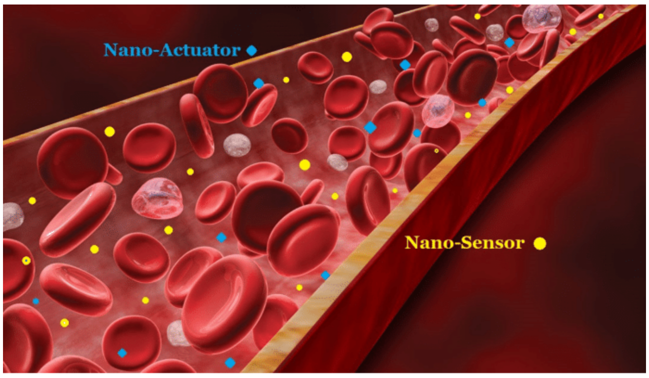


Fig. 2 – Drug delivery in human body.

This heterogeneity of the human body poses a major challenge to the routing protocol by traversing areas of different densities.

2.2 Agriculture sector

Agriculture is an important source of livelihood in most parts of the world. Wireless nano-sensor(s) have been used in modern agriculture and farming (Precision Agriculture (PA)), defined as the techniques of applying farming parameters and resources for production optimization, and reducing human effort [10].

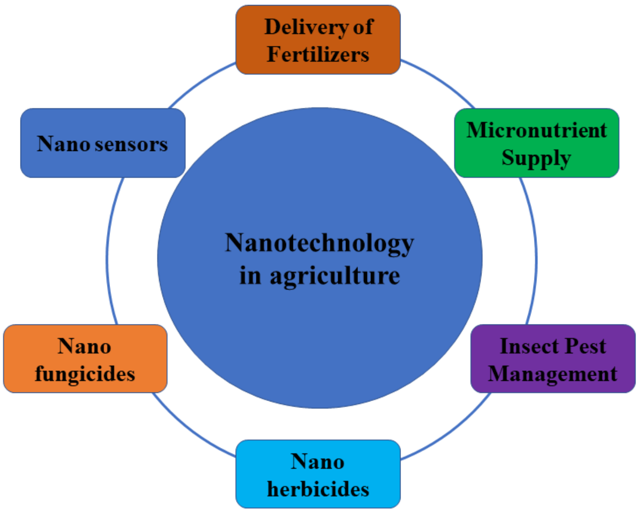


Fig. 3 – Nanonetwork technology impact in agriculture.

To improve productivity (e.g. detection of plant viruses, soil humidity degree), a farmer needs to monitor numerous parameters (Fig. 3). Wireless nano-sensors need to be used to realize the vision of intelligent agriculture. One can deploy autonomous nano-sensors around the plants to monitor soil condition and plant viruses, which gives them all the information about the environmental condi- tions of the plants using a simple portable device.

WNSN technology will contribute in generating the tools for establishing a real-time plant monitoring system, composed of a chemical nano-sensor merged with plants. Chemical nano-sensor nodes are miniaturized machines that interact with the environment to collect chemical compounds disseminated by plants. Micro-gateways can interconnect data collected from the nanonetwork to the external network, and to the decision officer of the analytical laboratory [11].

2.3 Civil engineering sector

Developments in nanotechnology can benefit construction engineering by enabling the practical deployment of structural condition monitoring systems for large civil engineering systems (Fig. 4). Nanonodes can be integrated into a composite material that can provide information on its performance and environmental conditions by monitoring structural loads, temperature, and humidity. It can be used for the construction of smart buildings [12]. Also, they could be coated into bridges, as prevention monitoring against degradation and cracking in order to alert the decision-making authorities well before the damage is de- tectable by human inspectors [13].

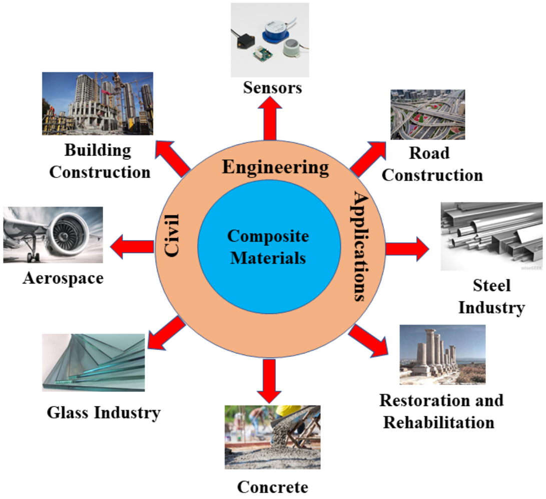


Fig. 4 – Smart composite materials based on nanotechnology, for civil engineering applications.

Nanonodes could also be embedded into roads and structures to allow engineers to monitor deterioration and cracking, thus saving physical intervention. Road sensor networks could gather and provide data to transport operators to better control and detect congestion and incidents.

2.4 Environment sector

Environmental safety is one of the sectors where nanotechnology may have a major impact. Through the installation of nano-sensors in high density public locations (e.g. hospitals, airports, and restaurants), authorities can trace the circulation of viruses and notice how various types of people are affected. Nanosensor networks could also be utilised to monitor the environment, such as pollution and gas emission.

Water is the lifeblood of the world, hence the importance of monitoring its quality and safety. We can benefit from nano-sensors in detecting bacteria, diseases and other harmful infectious agents, especially when improvement is made close to the point of use at the household. Nanosensors can also be used in flood-prone rivers near residential places. These sensors can measure the water level in rivers, and send the information to the control and decision room in real time (Fig. 5).

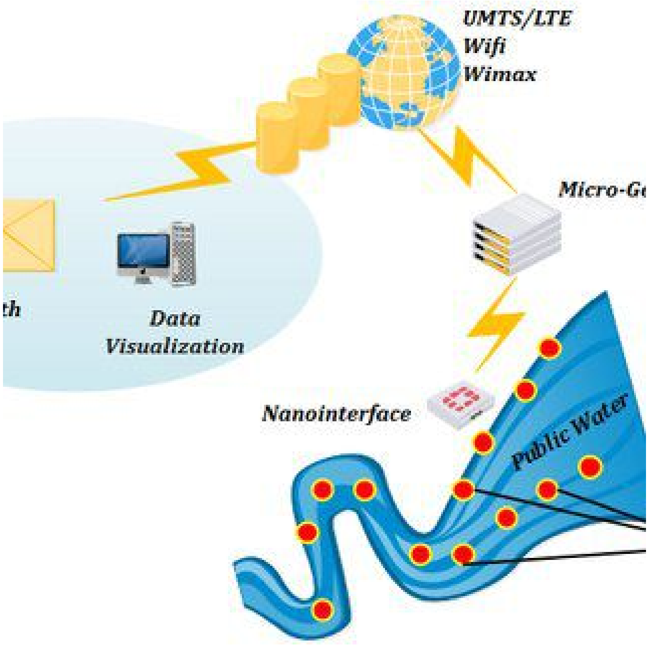


Fig. 5 – A water monitoring architecture.

To summarise, numerous applications use heterogeneous wireless networks. The biggest challenge remaining is the design of the routing protocols, which will face the challenges of different node densities in several regions, and in this context we focus on the influence of the sleep mechanism on the delivery of data packets to their desired destination. This is the focus of the evaluation section in this article.

3. BACKGROUND

In this work, we target potentially very dense, multi-hop nano-network(s). The contributions of this paper are presented later, but they are built upon a collection of protocols already independently designed for those dense nano-networks. In this section we give the necessary background concerning these protocols that we are using together: the routing protocol (SLR), the nano-sleep mechanism itself, and the density estimation protocol (DEDeN).

It is important to note that the choice of the routing protocol here is mostly inconsequential to the contribution. In this paper we do not present a new routing protocol. In regard to the very specific concurrent transmission behaviour of wireless nano-network(s), we instead present a mechanism to reduce the load on each node. This mechanism can be adapted to various routing protocols and here we choose SLR for the sake of convenience.

3.1 SLR protocol

Stateless Linear-path Routing (SLR) is a spatial routing/addressing protocol. It implements a coordinate-based routing, in which data packets are routed in a linear routing path. Coordinates are defined as an integer number of hops from special nodes called anchors (Fig. 6). All nanonodes are assumed to be placed in a cubic space, where anchors' nodes are placed at the vertexes (two anchors for a 2D area / three anchors for 3D area).

The SLR protocol has two phases: (1) addressing/initial phase and (2) routing phase.

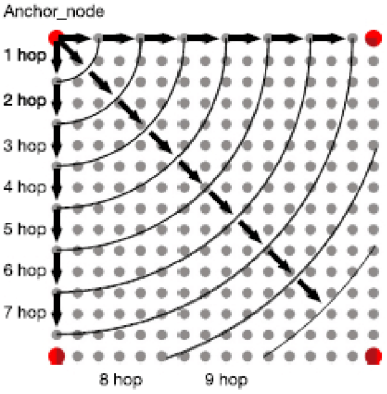


Fig. 6 – SLR addressing phase.

In the initial phase, addressing, as shown in Fig. 7, two anchor nodes (placed at the vertexes of 2D network) broadcast a packet (beacon) to the whole network. Beacon mes- sages include a hop distance field initialized to 0, which increments with each retransmission. The node coordinates are the hop counter in those beacons and correspond to the distance to the anchors. This phase will be performed only once at the network deployment.

In the routing phase (Fig. 7), packets contain the SLR address of the sender and receiver. A greedy approach is used to route the packets to their intended destination. When a node receives a message, it checks if it is located on the path from the source to the destination. If and only if it is the case, the node forwards the message. This check uses basic integer computation appropriate for the low nanonode computation capabilities.

Note that because of the way the coordinates are defined,they can be shared by multiple nodes (which belongs to the same SLR zone). Consequently, SLR is a protocol that routes a packet to the correct zone instead of a specific node.

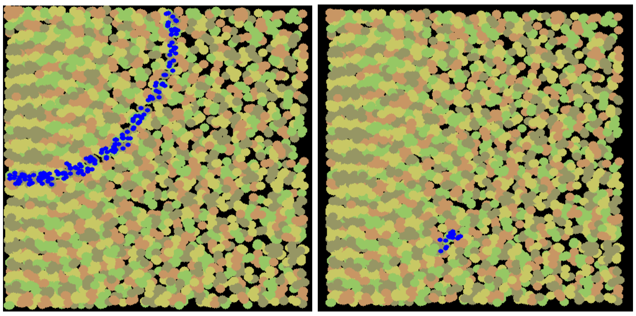


Fig. 7 – SLR initial phase (left) and SLR routing phase (right).

3.2 Sleeping mechanism

We start be emphasizing that the **nano-sleep** we present in this section is different in both means and objectives from the sleep mechanism (often call **duty-cycle**) that we commonly find in traditional networks.

In traditional networks, making a node sleep and wake up in defined intervals aims at preserving energy and thus increasing the life span of the network. A node always wakes up for a duration long enough to allow the reception of one or more frames. It processes the received packets and then goes back to sleep by turning off most of its reception and processing capabilities. By doing so, it preserves a substantial amount of energy. As the packets are sent sequentially over the channel, the longer the awaken duration, the larger the number of packets received. We could also say that while it is awake, a node has to be able to process the full throughput of the network. It then completely ignores packets sent while it is sleeping.

Our proposed nano-sleeping mechanism differs from those used in a macro-scale network by its fine granularity, asynchronicity, and decentralization. The main problem it solves is the potential overwhelming of a node when too many packets coexist on the channel. With TS-OOK, multiple nodes targeting different receivers can indeed transmit at the same time, and there is no hard limit to how many packets can **concurrently** coexist on the channel. Because of that, using a traditional awaken duration may cause a node to be overwhelmed when the number of packets concurrently transmitted exceed its decoding capabilities. Nano-sleeping is designed to tackle this problem, by allowing a node to concentrate on a small number of packets while completely (and safely) discarding the others.

A nanonode does not stay awake for the duration of one or several packets, but for a much shorter duration, a fraction of the T_s value. In our proposed sleeping mechanism, all nodes have the same awake-sleep cycle, equal to T_s. Inside the cycle, all the nodes have the same

awake duration (or percentage of T_s) , but the beginning of the awake interval is different for each node, and is randomly deter- mined. The mechanism ensures that if a node is able to pick the first bit of a packet, it will be able to pick all the following ones. More details on this sleeping mechanism are described in [14].
Fig. 8 depicts an example, where receiver nodes Recv1, Recv2 and Recv3 wake up at different times, but for the same duration. Recv1 and Recv2 are able to pick the bits from flows 1 and 2, as they arrive when they are awake. Recv3 is able to pick bits only from flow 3.

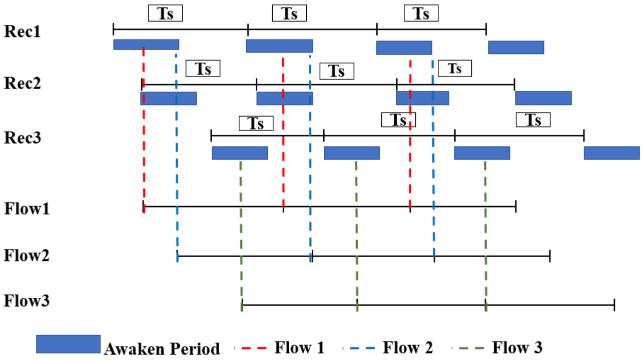


Fig. 8 – Sleep mechanism with three nodes and three flows.

The very high frequency of awake and sleep transitions make it significantly different from a traditional duty-cycle. Fully turning on and off processing capabilities at such a speed may not be technically achievable, but that is not the main point of our protocol anyway. Instead, we just consider that nodes do not process bits (pulses in TS-OOK) outside of their awaken duration.
As nodes do not turn off most of the hardware, the energy saving is limited (but still definitely exists, given that ignored frames are not processed and potentially retransmitted).
As we will see later, the benefit of this mechanism lies in an increase of the overall **effective** channel capacity. This benefit becomes really significant as the local node density increases and nodes can have individual lower and lower awaken durations.

3.3 Density estimator algorithm

Because nodes do not have much memory or processing power, elaborate routing protocols that try to find optimum forwarder(s) cannot work well if at all. Hopefully, we can improve routing protocols by using the local density of the network to limit their retransmission rate. We need a way for the nodes to discover by themselves how many neighbours they have.
A traditional solution to this problem is that each node sends "HELLO" packets. All nodes would only need to maintain a list of the neighbours from which they have received an HELLO packet to know the local density. This solution is simple and efficient if the density is low, but it becomes prohibitively costly in resources consumption as the density increases.

Density Estimator for Dense Networks (DEDeN) is an algorithm tailored to wireless nano-networks. It allows a node to estimate the number of its neighbours (also called node degree, node density, neighbour density, neighbourhood density, local density, or simply density).

DEDeN works by making nodes transmit an initialization message, and all nodes that receive it start the discovery process. With a very low (but known in advance) probability, they may start sending small beacons. As the probability is known, nodes receiving those beacons can easily compute an estimated local density. If the confidence in the estimation in not high enough, nodes start another round, with a higher beaconing probability. The confidence increases with the number of packets received and the probability to transmit.

The confidence and the error range of the estimation can be adjusted to the requirements of the user with a predictable overhead. Depending on how it is initiated, DEDeN enables a unique node or all nodes to estimate the neighbourhood density. The algorithm may be executed each time this neighbourhood density is needed.

For additional information about the density estimator algorithm, refer to [15].

4. RETRANSMISSION ALGORITHM AT THE DESTINATION ZONE

As explained in the previous section, the proposed sleeping mechanism should reduce the congestion and preserve node resources consumption. When a packet arrives at the destination zone, if the destination node is asleep, it will lose the data.

It is important to note that the proposed sleeping mechanism was implemented and tested in a homogeneous network. In this paper we are now evaluating this protocol in an heterogeneous network (areas of different nodes density). All nodes have the same awake-sleep time T_s, but they are different in their awaken duration and awaken starting time.

We propose an algorithm that increases the destination node chance of receiving a packet even if it was asleep when the packet reached the destination zone. The algorithm is to be used only by nodes at the destination zone. No matter the network density, it never saturates the radio channel and does not require much memory or computations. The only memory needed is the buffer to store the received packet to retransmit it at the end of the node awaken duration. A summary of the packet retransmission algorithm is presented in Algorithm 1.

Algorithm 1 Packet retransmission algorithm executed by all nodes at the destination zone except the destination node.

```
alreadyseen = false
waiting$T$ime
w$T$1ts = a$D$ - (pcktrecp - a$S$)mod $T_s$
w$T$2ts = - (a$D$ - (pcktrecp - a$S$))mod $T_s$
if packet type is data and reach the destination zone then
    if Sleeping mechanism is used then
        if packet !alreadyseen and the received node is not the destination node then
            if pcktrecp mod $T_s \geq$ a$S$ then
                waiting$T$ime = w$T$1ts
            else pcktrecp mod $T_s$ < a$S$ + a$D$ - $T_s$
                waiting$T$ime = w$T$2ts
            end if
        end if
    end if
end if
```

We recall that this algorithm works when the packet reaches the destination zone. The algorithm works on homogeneous and heterogeneous networks, with equal or different awaken durations. In this algorithm we took into consideration the case when the awaken duration spans over two time cycles. We describe the algorithm as follows:

- *waitingTime*: Represents the time the node should wait before packet retransmission at the end of its awaken duration.
- *wT1ts*: Formula to calculate the node waiting time if the awaken duration range is 1 T_s.
- *wT2ts*: Formula to calculate the node waiting time when its awaken duration spans for 2 T_s.
- *aD*: Represents the node awaken duration.
- *aS*: Represents the node awaken starting time.
- *pcktrecp*: Represents the time when the node receives the packet.

Fig. 9 depicts an example of a packet retransmission at the destination zone. In our proposed algorithm, all the nodes have the same awake-sleep cycle T_s. Inside the cy- cle, all the nodes are different in their awake duration aD (or percentage of T_s) , also the beginning of the awake in- terval for each node aS are randomly determined. When a node receives the packet, the algorithm checks whether the node is not the destination node. If this is the case, the algorithm will be applied.

Node 3 represents the destination node. Node 1 receives the packet, and since it is not the destination node, it executed the algorithm. The algorithm computes the waiting time to retransmit the packet at the end of the node awaken duration. By coincidence, nodes 2 and 4 are awake during the packet retransmission by Node 1. It happens that a node can receive several copies of the packet, which is the case for Node 4. This is due to the simultaneous awake of the node with the retransmission. Finally, the figure shows that the destination node awaken time does not overlap with any of the other nodes. Node 4 retransmits the packet at the end of his awaken duration, and it is received by the destination Node 3. Therefore, in the absence of the retransmission algorithm, the destination node would not have received the packet in the case where it was in sleep mode.

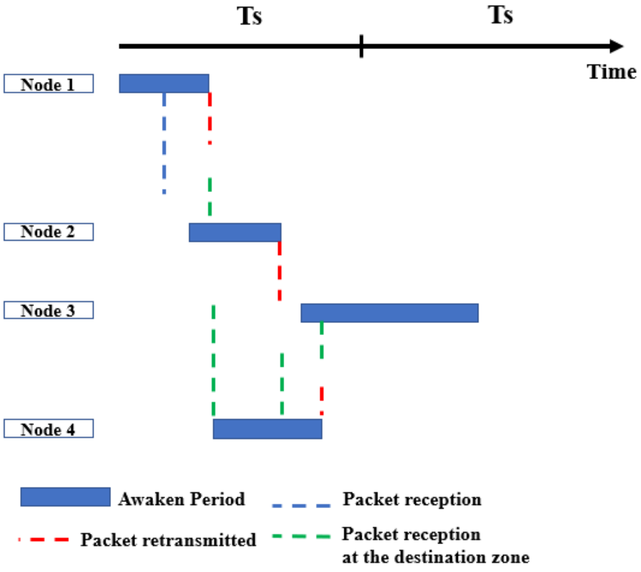


Fig. 9 – The effect of the retransmission algorithm at the destination zone.

5. EVALUATION

In this section, we evaluate our proposed ideas through simulations since a detailed analytical study is not possible. BitSimulator [16] is used in our simulations due to its characteristics.

BitSimulator has been designed to allow simulations of applications and routing protocols in wireless nanonetworks. It differs from other network simulators by its accuracy as it can simulate every bit of payload and their potential collisions. It also targets ultra-dense nanonetworks (hundreds of thousands of nodes). It is highly optimized for speed and accompanied by a VisualTracer, an interactive visualization and analysis tool that greatly helps in dense and complex scenarios. Technical details and information about full reproducibility of our results are provided on a separate website[1].

In our previous publication [14], we proposed a novel sleeping mechanism in a homogeneous nano-network which is totally different to the traditional sleeping mechanism applied in ad hoc networks. We define a homogeneous network as a network in which node density is constant. Therefore, all nodes will have the same awaken duration. In this kind of network, the nano-sleep mechanism

[1] \http://eugen.dedu.free.fr/bitsimulator

has been proven to work, and has contributed to improving the nodes' energy consumption as well as increasing network lifetime.

Nano-networks have different applications in various fields, extending from environment monitoring, industrial manufacturing, and agriculture to an enormous number of applications in medicine (like drug delivery, diagnostics and surgical operations).

Nowadays, the health sector is one of the most important sectors in which nanotechnology is involved. A nanonetwork can collect vital patient information and provide it to computing systems, providing more accurate and efficient health monitoring. These real applications will mostly be implemented in a heterogeneous nanonetwork due to its structural nature. This is why in this paper it was necessary to understand and validate the behaviour of the routing protocol used in places where nodes have different densities.

5.1 Network scenario

In our simulations, the network topology consists of a heterogeneous network as a 2D area of size 6 mm x 6 mm, shown in Fig. 10. It is composed of three rectangles of size 6 mm x 2 mm, with node average densities of: (rectangle 1: 160, rectangle 2: 80, and rectangle 3: 60). Nodes are always placed randomly except anchors, sources, and destinations nodes. Source nodes are placed at the bot- tom of the network near the edges.

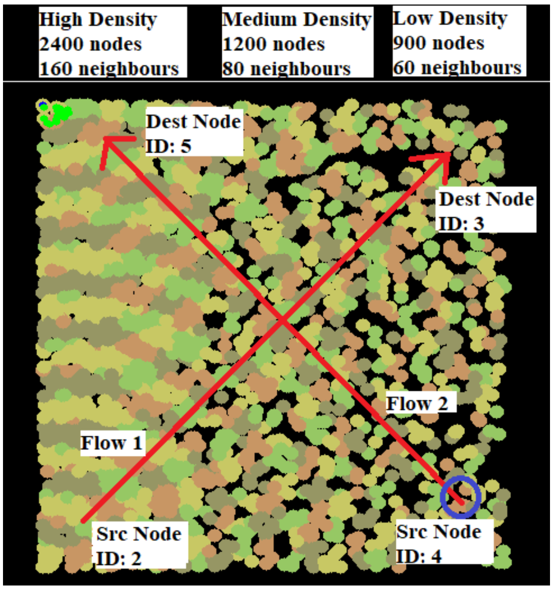


Fig. 10 – The network topology used, with 2 flows and 3 rectangles of different densities.

Given the environment size and the communication range (0.5 mm), the furthest possible receiver is in the corner at $\sqrt{2} * rectangle\ side/2 = 4.2$ mm away from the source. It means that a packet from the source needs at least 17 hops ($(4.2/0.5) * 2 = 17$) to reach the furthest possible node. Table 1 lists the parameters used in the simulations.

Table 1 – Simulation parameters

Size of simulated network	6 mm * 6 mm
Size of rectangles	6 mm * 2 mm
Number of nodes	4500
Communication radius	500 µm
Hops to reach the furthest node	17
β (time spreading ratio)	1000
T_p	100 fs

There are 2 flows. The source nodes are at the bottom right and bottom left of the network, while the destination nodes are at the top left and top right. Flow 1 traverses the network from left to right through different nodes' density areas (high, then medium, then low), while is the opposite for flow 2. According to the previous network settings, packet routing will be somewhat complicated and challenging.

The sleeping mechanism is applied to all nodes. To avoid random effects, each point in the following figures represents the average of 10 simulations, each with a different random generator seed to set the beginning of the sleeping period for nodes. In each simulation round, the awaken duration percentage and the number of neighbours are changed, while the other parameters are kept identical.

5.2 Comparison metrics

We analyse the sleeping mechanism efficiency using three metrics:

- *Packet reachability*: The number of packets that have reached the destination node. If several copies of the same packet reaches the destination, only one packet is counted.

- *Total number of sent packets*: The total number of sent packets in the network, for instance in a multihop transmission a same packet is sent by several nodes (routers), hence it is counted several times.

- *Awaken duration*: The interval of time where the node is awake. After this duration, the node goes back to sleep for the rest of the duty cycle (T_s). In BitSimulator, through the command line we can specify the percentage of awaken duration as a variable, or the average density of neighbour nodes (*awakenNodes*) via DEDeN that calculates it automatically.

5.3 Same awaken duration for all nodes

We recall that the time between two consecutive bits is $T_s = T_p * \beta$.. To determine an awaken percentage for every node in the network (e.g. 20% is equivalent to 20 000 fs) means that all nodes will awake for this percentage in a time duration equal to T_s. Inside the cycle, all the nodes have the same awaken duration (or a percentage of T_s), but the beginning of the awake interval is different for each node and is randomly determined.

Fig. 11 shows the average of packet reachability, i.e. at least one packet succeeding in reaching the destination node. We notice that packets are unable to reach the destination for an awaken duration ranging from 5 to 20%. This is due to the insufficient number of awaken nodes especially in low density areas, the flow propagation stops at the beginning. Recall that we are testing in a heterogeneous network, where areas' densities are different.

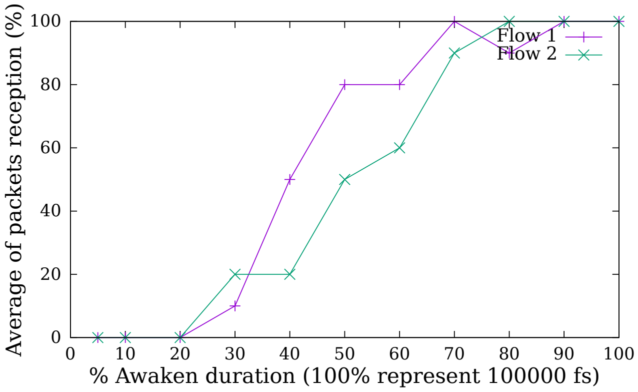


Fig. 11 – Percentage of arrived packets depending on awaken duration, for 2 flows.

Increasing the awaken duration immediately increases the average packet reachability. An awaken duration of 70 000 fs allows all packets from flow 1 to be received. Recalling that due to the sleeping phenomenon, it is possible for the packet to arrive while the destination node is asleep, and this results in the packet being lost. This is what the curve shows at 80 000 fs (flow 1). For flow 2, an awaken duration of 80 000 fs allows a total packet reception. *Note that 100% of awaken duration means that all nodes in the network are awake all the time.*

Given that the destination is far away from the source, the packet transmission between nodes occurs by hops, so a packet is sent several times, by several nodes in the path. The number of packets sent varies with the node awaken duration. Fig. 12 shows that the number of packets sent is very close to 0 at the beginning for both flows, due to the low number of awake nodes. Increasing the awaken duration percentage raises the number of awaken nodes, which results in an increasing number of packets sent too. This number reaches a value of 238 (flow 1) and 179 (flow 2) where the awaken duration is 90%.

5.4 Different awaken durations for nodes, based on local density

The sleeping mechanism dispatches the load of sent packets among neighbouring nodes. Therefore, taking into account the neighbouring nodes' density to determine the nodes awaken period can be beneficial to the transmission process. Fig. 13 shows a disparity in the nodes' awaken duration range. For an average density of 60 neighbours, the node awaken duration ranges between 30 000 fs and 100 000 fs. For a density of 100 neighbours, the node awaken duration range starts at 50 000 fs. As the average of neighbours' nodes increases, the awaken duration range shrinks and goes close to 100 000 fs.

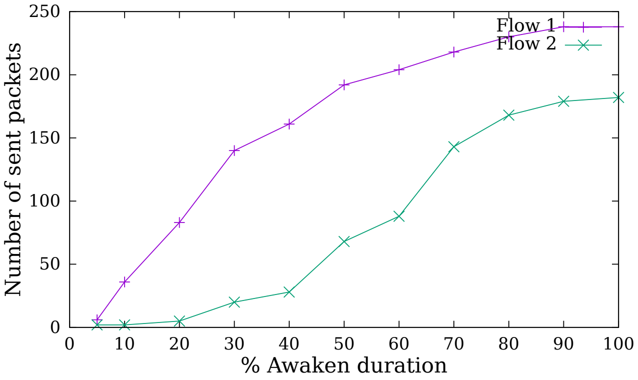


Fig. 12 – The number of packets sent varies with node awaken duration.

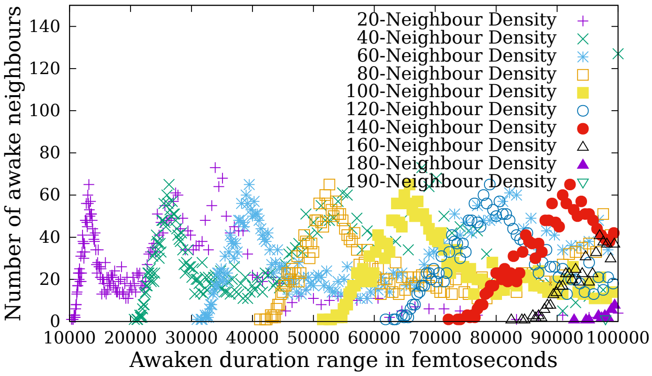


Fig. 13 – Number of awaken nodes in a duration range for different average densities.

Packet reachability can be significantly improved depending on the average density. For example, when specifying awakenNodes=20 to BitSimulator, each node executes a node density estimation algorithm and computes the awake interval according to a simple formula. Starting from 20 as average density, packet reception starts increasing (Fig. 14). An average density of 50 neighbours is sufficient for packets to reach their destinations for flow 1, whilst 60 neighbours for flow 2 are needed. Sleep- ing improves network behaviour by limiting the amount of traffic an individual node can see, but provisions have to be made to ensure the destination node is not sleeping when data packets reach it (example flow 2 at 130 neigh- bours nodes, where the curve has lost its stability). *Note that, in this scenario, an average density of 190 means that all nodes are awake all the time.*

When average density is low, most packets are not transmitted (because packet propagation stops at the beginning or in their path to the destination). Fig. 15 shows that for low average densities (e.g. 20, 30), the number of packets sent is quite low (close to 0). Indeed, with the increasing of the neighbours' nodes average density, packets sent will exponentially increases. For 140 neighbours, the number reaches its highest value at 238

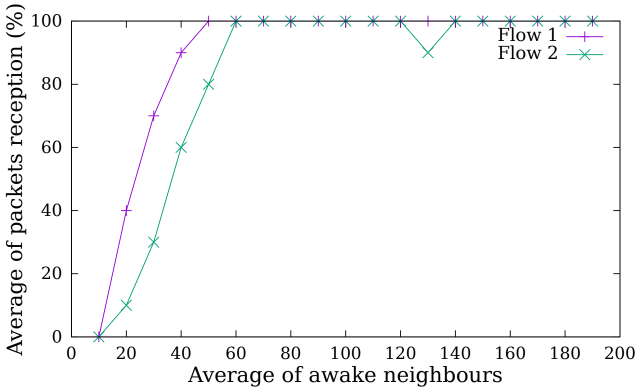


Fig. 14 – Percentage of arrived packets depending on neighbours nodes average, for 2 flows.

for flow 1 and 182 for flow 2. We notice that with 60 neighbours' nodes (*where all packets reach their destination*), the total number of packets sent for flow 1 is 205 and 155 for flow 2.

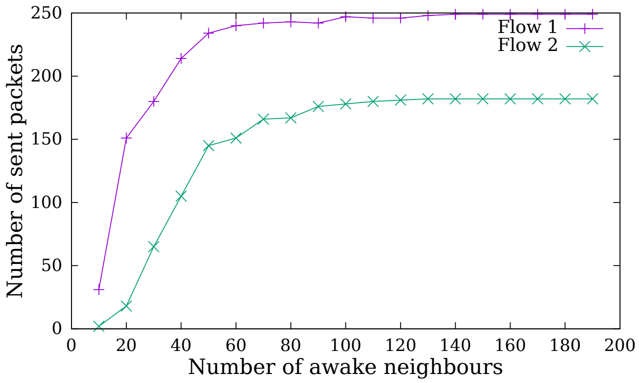


Fig. 15 – Packet transmission cost depending on the average of neighbours' nodes.

5.5 Processing at the destination zone

As explained before, the proposed sleeping mechanism reduces the congestion problem and preserves node resource consumption along the path from the source to the destination. But the very definition of destination may change depending on the application. If the destination is defined as an SLR address, it means that we want the packet to reach this SLR zone and that at least one node in this zone must receive this packet. In that case, the mechanism we proposed is efficient and can be used as is. On the other hand, if we aim to reach a specific node at the destination zone, then more aspects have to be taken into consideration. When a packet arrives at the destination zone, the destination node by chance may be asleep and consequently misses the packet. Different strategies might be used to solve this problem, depending on the node peculiarities, the local density and the application requirement. The algorithm is well explained in Section 4.

Fig. 16 shows the impact of applying our proposed retransmission algorithm to all the nodes at the destination zone. The algorithm clearly enhances the average of packet reception with the flow 1-patch compared to flow1-nopatch (without the algorithm). A value of 40 neighbours nodes ensures an average of 100% of packet reception by the destination node. Indeed, the positive impact is also clearly shown for flow 2-patch at the 130 of awake neighbours, comparing to the flow 2-nopatch, where the destination node missed the packet because it was asleep. The proposed algorithm shows its effect in increasing node chances in receiving the packet if it was asleep at the moment the packet reaches its zone (Fig. 17).

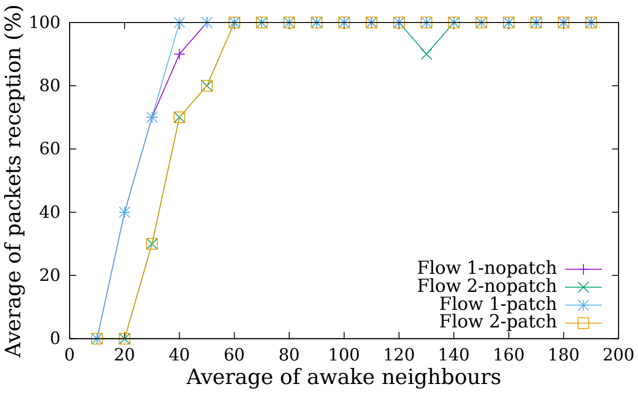


Fig. 16 – The impact of the retransmission algorithm.

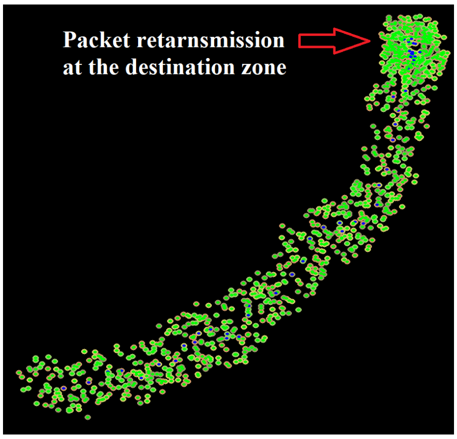


Fig. 17 – The retransmission algorithm applied at the destination zone increases the number of exchanged packets in that zone.

5.6 Average density vs awaken percentage

The node awaken interval, considered the main difference between the two ways of applying a sleeping mechanism. The DEDeN algorithm allows a node to estimate the number of its neighbours. Based on this value, nodes will set their awaken interval (*awaken duration*) (Fig. 18). However, fixing an awaken percentage (e.g. 80%), means that all the nodes in the network will be awake for the same duration (e.g. 80 000 fs).

The longer the node sleeps, the lower the consumed resources. The difference in nodes awaken duration is considered a special peculiarity of the average density (awakenNodes). An average density of 60 neighbours shows an awaken duration distribution (*less resource con-*

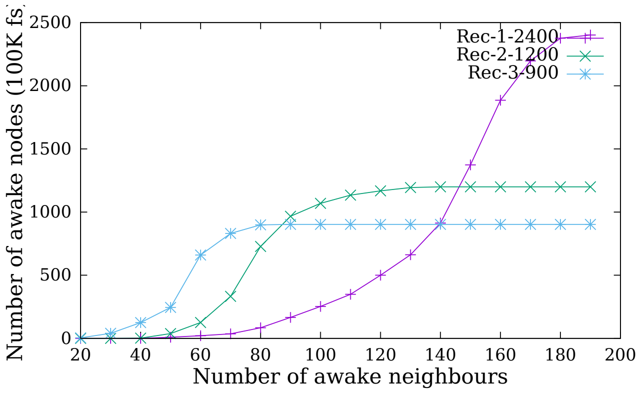


Fig. 18 – Number of full-awake nodes over different average density values.

sumption) ranging between 20 000 and 100 000 fs (*different awaken duration for each node*). Furthermore, for an awaken percentage of 90%, all the nodes in the network (4500 nodes) will be awake along this duration (Fig. 19).

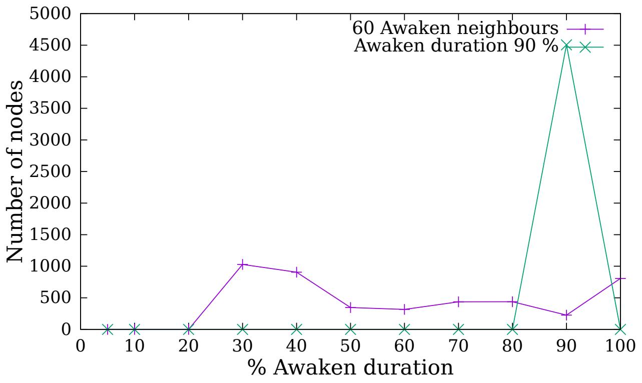


Fig. 19 – Number of nodes depending on an awaken duration of 90% and 60 neighbour nodes.

The total number of sent packets is a metric we rely on to compare between the methods used in applying the sleep mechanism (awakenDuration vs awakenNodes). Based on the results in the previous sections, setting the node awaken duration based on node neighbour density (using DEDeN) will help in decreasing the number of packets sent by 14% when compared to a fixed awaken percentage (awakenDuration).

To conclude, in heterogeneous networks, the SLR routing protocol is acting well when crossing areas of several densities. It smoothly deals with nodes of different awaken durations, and packets are successfully delivered to their intended destination. Applying DEDeN allows nodes to have different awaken durations compared to a static awakenDuration. This leads to a decrease in packet transmission cost, and preservation of node re- sources (CPU, energy, memory), therefore an increase in network lifetime. Moreover, applying the retransmission algorithm at the destination zone increases the chances for the destination node to receive its intended packet. Therefore, this algorithm helps to increase the reliability of data arrival to the destination node, which is an important parameter in some applications.

6. CONCLUSION AND FUTURE WORK

This paper presents the effect of the sleeping mechanism in a heterogeneous nano-network. The new idea embeds a density estimator algorithm (DEDeN) to automatically tune the node awaken interval (awakenDuration). The estimator algorithm can be executed in different times (network deployment, or when a node needs to estimate its number of neighbours). It shows its usefulness when used in conjunction with the sleeping mechanism. It becomes clear that relying on average neighbour node density has a positive influence in preserving node resources (CPU, energy, memory ...) and decreases the number of sent packets compared to results based on static percentage of awaken duration. Considerations have also been taken for the specific case of the destination zone (packet loss by the destination node because its sleep may coincide with the packet's arrival at the destination zone). An algorithm has been proposed in case the application needs data packets to reach a specific node in the destination zone. As future work, we would like to consider different algorithms and variations to handle what happens at the destination zone. Especially, a destination could be expressed as "any node in the destination area that provides a given service" instead of all nodes in the zone or a single, specific node.

7. ACKNOWLEDGMENTS

Ali Medlej has a grant from Islamic Center Association for Guidance and Higher Education (Lebanon). This work has also been funded by Pays de Montbéliard Agglomération (France).

REFERENCES

[1] Xin-Wei Yao and Ye-Chen-Ge Wu and Wei Huang, ”Routing techniques in wireless nanonetworks: A survey”, in Nano Communication Networks, pp. 1–13, Sep, 2019, Elsevier.

[2] Büther, Florian and Lau, Florian-Lennert and Stelzner, Marc and Ebers, Sebastian, ”A formal definition for nanorobots and nanonetworks”, in Internet of Things, Smart Spaces, and Next Generation Networks and Systems, pp. 214–226, Aug, 2017, Springer.

[3] Akyildiz, Ian F and Jornet, Josep Miquel, ”The Internet of nano-things”, in IEEE Wireless Communications, pp. 58–63, Dec, 2010.

[4] Jornet, Josep Miquel and Akyildiz, Ian F, ”Femtosecond-long pulse-based modulation for terahertz band communication in nanonetworks”, in IEEE Transactions on Communications, pp. 1742–1754, Apr, 2014.

[5] Tsioliaridou, Ageliki and Liaskos, Christos and Dedu, Eugen and Ioannidis, Sotiris, ”Packet routing in 3D

nanonetworks: A lightweight, linear-path scheme", in Nano Communication Networks, pp. 63–71, Jun, 2017, Elsevier.

[6] Akyildiz, Ian F and Jornet, Josep Miquel, "Electromagnetic wireless nanosensor networks", in Nano Communication Networks, pp. 3–19, March, 2010, Elsevier.

[7] N. Ali and M. Abu-Elkheir, "Internet of nano-things healthcare applications: Requirements, opportunities, and challenges", in IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Los Alamitos, CA, USA, pp. 9–14, Oct, 2015.

[8] Maksimovi , Mirjana, "The roles of Nanotechnology and Internet of Nano things in healthcare transformation", in TecnoLógicas, pp. 139–153, Dec, 2017.

[9] Seyedi, MirHojjat and Kibret, Behailu and Lai, Daniel TH and Faulkner, Micheal, "A survey on intrabody communications for body area network applications", in IEEE Transactions on Biomedical Engineering, pp. 2067–2079, Aug, 2013.

[10] Lee, Won-Suk and Alchanatis, Victor and Yang, Chenghai and Hirafuji, Masayuki and Moshou, D and Li, Changying, "Sensing technologies for precision specialty crop production", in Computers and electronics in agriculture, pp. 2–33, Oct, 2010, Elsevier.

[11] Giraldo, Juan Pablo and Wu, Honghong and Newkirk, Gregory Michael and Kruss, Sebastian, "Nanobiotechnology approaches for engineering smart plant sensors", in Nature nanotechnology, pp. 541–553, Jun, 2019, Nature Publishing Group.

[12] Wegner, Theodore H and Winandy, Jerrold E and Ritter, Michael A, "Nanotechnology opportunities in residential and non-residential construction", in NICOM 2: 2nd International Symposium on Nanotechnology in Construction, pp. 339–347, Nov, 2006, RILEM Publications SARL.

[13] Zheng, Wei and Shih, Hui-Ru and Lozano, Karen and Mo, Yi-Lung, "Impact of nanotechnology on future civil engineering practice and its reflection in current civil engineering education", in Journal of Professional Issues in Engineering Education and Practice, pp. 162–173, Jul, 2011, American Society of Civil Engineers.

[14] Ali Medlej and Kamal Beydoun and Eugen Dedu and Dominique Dhoutaut, "Scaling up Routing in Nanonetworks with Asynchronous Node Sleeping", in 28th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Hvar, Croatia, pp. 177–182, Sep, 2020, IEEE.

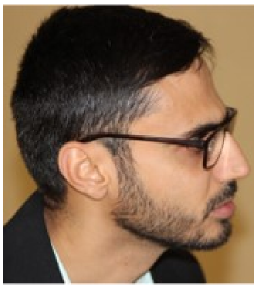
[15] Arrabal, Thierry and Dhoutaut, Dominique and Dedu, Eugen, "Efficient density estimation algorithm for ultra dense wireless networks", in 27th International Conference on Computer Communication and Networks (ICCCN), pp. 1–9, Jul, 2018, IEEE.

[16] Dominique Dhoutaut and Thierry Arrabal and Eugen Dedu, "BitSimulator, an electromagnetic nanonetworks simulator", in 5th ACM/IEEE International Conference on Nanoscale Computing and Communication (NanoCom), New York, NY, USA, pp. 1–6, Sep, 2018, Association for Computing Machinery.

AUTHORS

A. Medlej holds a degree in computer and telecommunications engineering and a specialized master's in telecom, network and security, option: cybersecurity engineering in (2018) from the Lebanese University – Faculty of Engineering and Saint Joseph University of Beirut-Beirut Higher School of Engineering. He starts a co-direction PhD between Franche-Comté University, SPIM Doctoral School, FEMTO-ST lab - Nano-Network Research Team, Montbéliard, France and the Lebanese University, EDST Doctoral School, Faculty of Sciences I, L'ARICoD lab, Hadath, Lebanon, entitled « Congestion control in dense nano-networks» under the direction of Doctor Eugen Dedu (Franche Comte) and the supervision of Doctor Kamal Beydoun (Lebanese University) and Doctor Dominique Dhoutaut (Franche Comte).

E. Dedu has been an associate professor at the University of Franche-Comté (UFC) in Montbéliard, France since 2003. He is a member of the Department of Computer Science and Complex Systems (DISC) of the FEMTO-ST laboratory. He received his Ph.D. in Computer Science at University of Versailles St Quentin, France, in 2003. He received an engineering degree from University Politehnica in Bucarest, Romania, in 1998. He co-authored around 50 articles in international journals and conferences. His current research focuses on wireless dense networks and wireless nanonetworks.



K. Beydoun is an associate professor at the Lebanese University (LU), Beirut, Lebanon, and a member of the Faculty of Sciences since 2011. He obtained his undergraduate degree at the Lebanese University (2001) in computer science. Beydoun received his Ph.D. in computer net- working at Franche-Comte University, Besancon, France under the supervision of Professor Hervé Guyennet. A fellow researcher in LARICoD Laboratory at the Lebanese University collaborating with international research labs and universities, his current research focuses on wireless dense networks and wireless nanonetworks. His work is situated in the crossdisciplinary study of computer science, particularly Wireless Sensor Networks(WSN) and Intrusion Detection System(IDS). Kamal teaches several undergraduate courses in software engineering and programming languages.

D. Dhoutaut is an associate professor at the University of Franche-Comté (UFC) in Montbéliard, France. He is a member of the Department of Computer Science and Complex Systems (DISC) of the FEMTO-ST laboratory. After a PhD on mobile ad hoc networks and IEEE 802.11 at INSA Lyon and also consecutive work on Vehicular Ad Hoc Networks, his research interests now reside in nano-networks. He develops protocols at various layers and also specializes in evaluation tools, such as complex networks simulations software.