EIDA, a Best Effort Equitable Distributed id Assignment Mechanism for Heterogeneous Dense Nanonetworks

Carole Al Mawla
Université de Franche-Comté, CNRS, institut FEMTO-ST
Montbéliard, France
carole.almawla@univ-fcomte.fr

Eugen Dedu
Université de Franche-Comté, CNRS, institut FEMTO-ST
Montbéliard, France
eugen.dedu@univ-fcomte.fr

Dominique Dhoutaut
Université de Franche-Comté, CNRS, institut FEMTO-ST
Montbéliard, France
dominique.dhoutaut@univ-fcomte.fr

Kamal Beydoun
L'ARICoD Laboratory, Faculty of Sciences I, Lebanese University
Beirut, Lebanon
Kamal.Beydoun@ul.edu.lb

ABSTRACT

Assigning identifiers (ids) in dense nanonetworks is challenging due to the large number of nodes, the potential for interference and signal overlap, and difficulties in ensuring fair and efficient id assignment. Existing solutions, such as ideal and random assignments, have limitations and are unsuitable. To address these issues, we propose EIDA, a configurable distributed mechanism that assigns ids in the best effort of equitability. EIDA combines ideal and random assignments, where some nodes assign their ids using packet exchanges until a minimum guarantee is fulfilled, and the remaining nodes assign ids randomly without further communication. Our simulations on a ultra-dense nanonetwork demonstrate the effectiveness of EIDA in assigning ids. Our mechanism allows routing protocols, for instance, to select a desired number of nodes to forward at each hop.

CCS CONCEPTS

• **Networks** → *Network algorithms*; *Network protocols.*

KEYWORDS

Dense nanonetworks, id assignment problem.

ACM Reference Format:
Carole Al Mawla, Eugen Dedu, Dominique Dhoutaut, and Kamal Beydoun. 2023. EIDA, a Best Effort Equitable Distributed id Assignment Mechanism for Heterogeneous Dense Nanonetworks. In *10th ACM International Conference on Nanoscale Computing and Communication (NANOCOM '23), September 20–22, 2023, Coventry, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3576781.3608715

NANOCOM '23, September 20–22, 2023, Coventry, United Kingdom

ACM ISBN 979-8-4007-0034-7/23/09…$15.00
https://doi.org/10.1145/3576781.3608715

1 INTRODUCTION

Dense multi-hop networks are composed of numerous nodes with high density, i.e. numerous neighbors. These networks present significant challenges for efficient communication, requiring unique node identification (id) as a fundamental primitive for several operations. One such operation is multi-hop routing, where packets are relayed through multiple nodes to reach their destination. However, without proper routing mechanisms, this process can be highly inefficient and can cause energy waste.

To avoid this inefficiency, nodes can be partitioned into zones, with each zone having a designated cluster head (only one node) responsible for forwarding packets. This approach can significantly improve network efficiency by reducing the number of nodes involved in packet forwarding. However, zones need to elect their cluster head and have very high load unbalancing, and these nodes deplete their battery very fast.

Another method is to partition nodes into groups, and hence split traffic among potential forwarders, where each packet is processed by only one group of nodes. To achieve equitable traffic splitting, i.e. avoiding that one flow is processed by e.g. 18 nodes and the other by only 2 nodes, groups need to have equal sizes. One way to achieve this is by assigning a unique id to each node and then using a simple modulo operation (id divided by the number of groups) to divide them into groups. FR-SLR [1] is an example of routing protocol, where only nodes with specific id participate in the routing process along the transmission path, and EIDA can be used to assign ids to nodes.

A nanonetwork consists of a set of interconnected nanomachines, devices that are a few micrometers at most in size. They are able to perform only very simple tasks such as computing, data storing, sensing, and actuation. It enables new applications of nanotechnology in the biomedical field, environmental research, military technology, and industrial and consumer goods applications. Nanonetworks currently use the TS-OOK (Time Spread On-Off Keying) modulation, which is based on femtosecond-long pulses in the terahertz band, appropriate for the very limited energy of nanonodes [4]. Bits are sent using a sequence of pulses interleaved by a randomly selected constant duration.

This paper proposes a method of id assignment in dense networks, in particular in nanonetworks, where nodes are very small and limited in memory storage, energy, and computing capabilities.

Designing such a method is challenging due to the large number of nodes, potential interference and signal overlap, and difficulties in ensuring fair and efficient assignments.

Existing assignment methods have drawbacks. For example, the classical ideal assignment, where each node receives consecutive ids (0, 1, ..., $n-1$), requires numerous packet exchanges among the nodes, which is unacceptable in a dense nanonetwork. The random assignment does not guarantee equitable distribution of ids, i.e. two groups may have 5 nodes while the third one has no node, leading to routing failures (die-out) when the latter group is used for routing.

To tackle this problem, we propose EIDA (Equitable Id Assignment), a distributed algorithm to assign ids in a best-effort equitable way. EIDA combines ideal and random assignments, where some nodes assign their ids using packet exchanges (like in the ideal assignment) until the required minimum guarantee is fulfilled, and the remaining nodes assign ids randomly without further communication. It works in one-hop nanonetworks, and in multi-hop nanonetworks after division into several independent zones processed separately by EIDA. It assigns ids based on the node density and two configurable parameters: redundancy and guarantee. While the network density is usually a fact and is fixed, the two parameters can be chosen by the user based on the application's needs. The *redundancy* represents the expected (desired) number of nodes in a group, and the *guarantee* represents the minimum ensured number of nodes in each group (e.g. we need at least 2 nodes in each group).

The evaluation of id assignment is done on electromagnetic nanonetworks, made up of tiny nodes of nanometric size.

The contribution of this paper is twofold:

- We introduce EIDA, a novel best-effort equitable id assignment mechanism.
- Simulations show a good trade-off between the amount of packet exchanged and the equitability while ensuring a minimum number of nodes in each group, as compared with alternative methods.

The remainder of this paper is organized as follows. Section 2 presents the related work on id assignment methods. Section 3 presents the assumptions, formalization, and description of the proposed EIDA mechanism. Section 4 shows how EIDA can be applied to multi-hop nanonetworks and the motivation for using it. Section 5 evaluates EIDA. Finally, the conclusion is drawn in section 6.

2 RELATED WORK

This section first explains the random and ideal assignment mechanisms on which our method is based and then proceeds with other methods. The methods are compared with our EIDA mechanism with respect to the number of packets exchanged (an important parameter in our case of ultra-dense nanonetworks), the guarantee of minimum group size, and the equitability of group sizes.

2.1 Random assignment

Random assignment is a simple mechanism where each node in the network chooses a random number as its id. These numbers are generated by each node in a distributed manner using an RNG (random number generator) [8] with different seeds on each node, i.e. there is no centralized node such as the base station in telecommunication networks that generates different ids for each node.

Usually, an RNG is used to generate random or unpredictable numbers. Instead, in this case, the RNG is used on a large number space to get *non-duplicate numbers*.

The main advantage of the random assignment mechanism is that no packet exchange (communication) is needed for nodes to choose their ids. Conversely, the major drawback is that it is not equitable, since there is no guarantee that nodes will be approximately equally distributed into the groups (when groups are small).

2.2 Ideal assignment

In the ideal assignment, nodes assign alternatively a unique id between 0 and $n-1$ (0, 1, 2, ..., $n-1$). This assignment is considered ideal since node ids are uniformly distributed without requiring a large space number.

One way to achieve this ideal assignment is to start with a random assignment, where each node chooses a random number within an interval much bigger than n to avoid two nodes choosing the same number. Nodes then transform these random numbers into ids by sending packets containing their number using a high backoff to avoid packet collisions. Once all the packets have been sent, all nodes know all the numbers, sort them, and get their ids. For example, if the randomly generated numbers are 58, 23, and 88, nodes will replace them with ids 1, 0, and 2, respectively.

The advantage of the ideal assignment is that it is 100% equitable. For example, using modulo operation generates the same number of nodes in each group (± 1). Its drawback is that it requires (generates) n packet exchanges, where n denotes the number of nodes.

The method presented in this article is a trade-off between random and ideal assignments, where the number of packets exchanges required is much smaller than n, while still providing a configurable guarantee on minimum group size.

2.3 Other assignment methods

Other id assignment methods include: during the manufacturing process, using tree data structures, and centralized approaches. We explain them and provide their unsuitability for our case below.

Ids can be assigned during the manufacturing process [12], using a globally unique id per node, such as the MAC address in traditional networks. This approach requires very long ids (to make them globally unique) and extra steps in the manufacturing process. Otherwise, it is equivalent to random assignment in a big space; hence, it has the same advantages and drawbacks.

[2] studies the unique node id assignment in modular robots. First, it builds a tree of nodes with a leader and calculates the number of children of every node (nodes below it in the tree). Then, a message is sent at each level, from parent to its children, based on the number of children already calculated. This process is repeated from parent to child until all modules in the system receive their globally unique ids. This mechanism leads to unique id assignments. The message exchange complexity of this algorithm is high, $O(n)$ (order of node density).

SIDA (self-organized id assignment) [6] is a local and distributed variable-length id assignment mechanism. The id space for the assignment expands in real-time when more nodes enter the network.

It builds an overlay binary tree in which each position is mapped to a unique id, which is then assigned to a node. Distant nodes from the sink receive shorter ids, while closer nodes are assigned longer ids. The algorithm runs in three phases: exchange control information between neighboring nodes, broadcast the maximal length to the sink node, and assign the unique ids of nodes. Each node sends 4 messages on average during these phases, which is too high.

[9] presents a distributed algorithm to assign unique ids using the minimum number of bytes. It has three phases: create the tree structure and assign temporary long ids, report the size of sub-trees from leaf nodes to the root (using the temporary ids), and assign the unique ids of nodes (each parent node assigns ids to its children nodes, starting from the root). By knowing the size of the network (using phase 2), the initiator can compute the minimum number of bytes required to assign a unique id to each node in the tree. This assignment requires numerous packets to be exchanged, for instance, it ends in about 5 minutes in a network of 1000 nodes [9].

The efficient topology discovery protocol, ETDP [14], assigns node ids using a tree structure. It establishes the layered network structure by transmitting topology discovery (TD) packets. Nodes are classified into different layers based on the TD packet exchange. The protocol begins with the root node sending a TD packet, and nodes in its communication range that receive it are considered its children. Each node in the network computes its unique id by exchanging id assignment packets with other nodes using network topology information, the father's id, a timer, the number of neighbors, and the layer number. This protocol requires more than $2 * O(n)$ packet exchanges.

[10] presents three algorithms: ids swapping assignment, probabilistic simulated annealing assignment, and distributed ids swapping assignment. The first two algorithms are centralized and appropriate for wireless sensor networks with fixed infrastructure, which is not our case; they exchange node ids between two nodes swapped if graph connection rules are respected. The last algorithm consists of four main steps: (1) announce its own node id to its physical neighbors, (2) collect candidate ids that want to be swapped, (3) select the best-fit node id to be swapped, and (4) swap node ids and update each logical neighbor table of its physical neighbors. It is distributed and efficient for ad hoc WSN. This algorithm generates unique but not ideal id assignments, with the same drawbacks as above.

It is worthwhile to note that [2, 6, 9, 14] generate unique (i.e. non-duplicate) ids, whereas the objective of our paper is to assign ids into different groups. To be able to use this assignment as in our case, we need to consider the number of groups. However, even when applying the modulo operation (%), the generated assignment might be unsatisfactory. For example, using $g = 2$ groups and unique ids 11, 33, 55, all three nodes are assigned to the same group (group 1), since $11\%2 = 33\%2 = 55\%2 = 1$.

3 EIDA IN ONE-HOP NETWORKS

EIDA is a method to assign ids to nodes in a best-effort equitable manner with a guarantee on the minimum. We present it in two steps: this section presents EIDA in a dense one-hop nanonetwork (i.e. each node is in the communication range of any other node), and the next section shows how it can be used in a multi-hop nanonetwork.

This section first outlines the assumptions used by EIDA. Afterward, it formalizes the proposed mechanism, describes it in detail, and finally considers its memory and energy requirements.

As a reminder, EIDA is a combination of the ideal and random assignments and solves the problem of the high number of packet exchanges of the first and the non-equity of the second.

EIDA has the following assumptions:

- Nodes do not appear or disappear in the network during the id assignment process.
- Nodes do not process concurrent packets, i.e. a node cannot send a packet while other packets are in the sending or receiving state. Indeed, some networks allow this type of concurrency, such as nanonetworks, and EIDA does not work if appropriate countermeasures are not taken (a high backoff in our case, as shown later).
- During the algorithm, packets reach the destination without loss.

3.1 Formalization

In a dense nanonetwork with n nodes, each node is within communication range of all the others. Our goal is to assign a unique id to each node, which allows us to assign every node to a group using a simple modulo operation (groupable id assignments). The number of groups g is computed based on a configurable parameter r (for redundancy, the expected number of nodes in each group), where $g = \lceil n/r \rceil$ (ceiling operation).

We want to assign each node to one of the g groups. Given that nodes are distributed, they are allowed to exchange packets.

We formalize this mechanism as follows: Let's assume we have a network of n nodes and a redundancy r. We search for a distributed method to divide the nodes into g groups, each with $r = \left[\frac{n}{g}\right]$ or $\left[\frac{n}{g}\right] + 1$ nodes, where the symbol $[X]$ denotes the integer part of X. Formally, the method should output $X(n, r, i)$ as the set of nodes in the group g_i so that:

$$|X(n, r, i)| = r \text{ or } r + 1, \quad i = 0, \ldots, g - 1 \tag{1}$$

where the symbol $|X|$ denotes the cardinality of the set X.

This distribution corresponds to the ideal assignment of ids to nodes (nodes are assigned to groups equitably). However, to achieve this distribution, all nodes need to exchange packets containing their ids (or the group they belong to). Consequently, this approach can be very costly in terms of resources, especially in a dense nanonetwork where n exchanged packets are required.

To avoid the huge amount of packet exchange required, we assume that some applications do not need an ideal assignment but only a minimum guarantee. Thus, we replace the constraint on r (equation 1) with another constraint, m, which specifies that each group must have *at least* m nodes (with $m \leq r$). This allows for an assignment as equitably as possible.

This new problem is formalized by the following new equation, which replaces equation 1:

$$|X(n, r, i)| \geq m, \quad i = 0, \ldots, g - 1 \tag{2}$$

For a better understanding, here is a numerical example:

- n = 100 nodes
- r = 5 redundancy, i.e. the *expected* number of nodes in each group
- $g = \lceil n/r \rceil$ = 20 groups
- m = 2 guarantee, i.e. the *minimum* number of nodes in each group.

We will see later that the number of packets needed to fulfill the constraint on the minimum is:

$$max\_pkts = n\frac{m}{r} \tag{3}$$

i.e. 40 packets, much smaller than $n = 100$ packets required to fulfill the constraint on full equity (for the ideal assignment).

3.2 EIDA functioning

Nodes can start the EIDA algorithm at the same time or at different times. In the latter case, when a node starts the algorithm, the packet it sends will "awake" the other nodes upon receipt, prompting them to start the algorithm as well.

Each node starts by choosing a random backoff in a very large window. The window is large in order to prevent two nodes from starting the algorithm simultaneously and to avoid two packets being received at the same time (because nanonetworks allow this). It is proportional to the number of nodes n, the packet size, and the time required for the packet to reach the receiving nodes.

During the backoff waiting period, when a node receives an EIDA packet, it increments its crt_id (id counter) value (representing the number of nodes having already assigned their ids), which is initially set to 0. Given that all nodes receive the packet (one-hop nanonetwork) and only one packet is sent at a time, all nodes will come up with the same crt_id value.

At backoff expiration, nodes compare the number of packets received with a threshold that is computed from the specified minimum guarantee. If smaller, it chooses its id as the value of crt_id and sends a packet to inform all the nodes of its choice of crt_id. Elsewhere, it chooses its id randomly without any communication, aiming for best-effort equitability.

Thus, the algorithm consists of two phases: phase 1, where nodes choose their ids incrementally (like in ideal assignment) and send a packet with their id, and phase 2, where nodes choose their ids using a random value (like in random assignment). As a minor point, in phase 2, instead of generating a new random id, each node can reuse the value of its chosen backoff since this value is already random.

Packets sent during the EIDA algorithm contain two fields: the specific type `EIDA` not to confound it with other packets in the network, and the id chosen by the sender.

Each node stops the algorithm at the end of its id assignment. Note that the communication between nodes stops when the threshold condition is fulfilled, i.e. when the maximum number of exchanging packets ($max\_pkts = mg = mn/r$) is reached, as explained above. At this point, all the nodes without assigned ids at their backoff choose their ids in the addressing space based on their backoff value.

The full algorithm is shown in Algorithm 1.

EIDA is designed to handle packet loss during the assignment process. If a node n_i sends a packet that gets lost, the algorithm continues as usual, and the other nodes continue at their backoff as planned. This ensures that the assignment process remains reliable and guarantees the same results as if there were no packet loss. The only difference is that node n_i will retain its id from phase 1 rather than phase 2, though it is still considered to be randomly assigned in phase 2.

Algorithm 1 ID assignment algorithm.

```
Input:
  n                          ▷ network density, number of nodes
  m    ▷ guarantee, minimum nb of nodes in each group, e.g. 2
  r          ▷ redundancy, expected nb of nodes in group, e.g. 5

g ← n/r                                   ▷ number of groups
crt_id ← 0                                ▷ next id to assign
max_pkts ← m * g          ▷ max number of packets exchanged
x ← rand()                                ▷ backoff value
wait (x)
if crt_id < max_pkts then                 ▷ phase 1
    id ← crt_id%g
    send_packet (EIDA, ids)
else                                      ▷ phase 2
    id ← x%g
end if

Upon packet reception:
if packet_type==EIDA then
    crt_id + +
end if
```

3.3 Memory and energy considerations

At the nanonetwork scale, memory and battery energy are severely limited [5].

The memory utilization of nodes is optimized in this algorithm since they only store the current ids, regardless of the network density.

The energy consumption depends, among others, on the number of exchanged packets, which is mn/r (cf. eq (3)). Note that only a subset of the nodes exchange packets, while others do not send any, so the energy consumption varies among nodes.

4 APPLICATION TO MULTI-HOP NETWORKS

For EIDA to work in a multi-hop nanonetwork, the network needs to be partitioned into disjoint zones and to apply EIDA in each zone separately. This is the case for networks divided into clusters (each cluster being a separate zone) or GPS (Global Positioning System) zones.

This section discusses the need for an id assignment mechanism in some applications. In addition, we present SLR [13], Stateless Linear-path Routing, a specific protocol used to divide the nanonetwork into zones and for routing. Afterward, we show how to apply EIDA in multi-hop nanonetworks using SLR.

4.1 Applications

This section presents the need for an id assignment mechanism in some applications and the motivation to use EIDA for these applications.

In a dense nanonetwork, where numerous nodes act as potential routers, node ids can be used to divide communication bandwidth or other resources, such as memory and energy, among the nodes. For example, without ids, nodes have to communicate by flooding the entire network, while adding the source and destination node ids to packets allows for efficient routing during communication by delivering the packet in a linear path from source to destination. EIDA can be used here to split traffic based on the ids (groups). Consequently, the packet is delivered in a linear path as well, but instead of having all the nodes in the zones (belonging to the traffic) retransmit the packet, only one group in each zone retransmits.

Other applications that use id assignment exist, including monitoring the temperature of a certain area where multiple nodes in the same area can hold the same ids (belong to the same group), object tracking, and location-aware ids where the id assignment of nodes depends on the coordinate system (hop count) like in phase 1 of SLR (section 4.2) [7].

A possible application in routing is FR-SLR zone splitting [1] in a multi-zone nanonetwork, where only some of the nodes in the zones of the path participate in the routing process. Each node has an id assigned by EIDA, and a modulo operation decides whether the node forwards the packet or not. For example, consider three nodes, A, B, and C, with ids 1, 1, and 0, respectively, where modulo 2 is used (number of groups) to decide the forwarding. When they receive the packet 0, only node C forwards it, since only id[C]%2 == 0. Using EIDA avoids transmission die-out as is the case when using random assignment that might assign ids 1, 1, and 1, respectively (this case can appear in $1/2^3 = 1/8$ of cases), so none of the nodes retransmits packet 0 (the modulo always returns 1). In FR-SLR, the parameters n_i, r and m refer to each zone. EIDA is applied independently in each zone. For any transmitted packet, we need $r \leq n_i$ retransmitters in each zone. The retransmitters must not be the same for all the packets but evenly distributed among the n_i nodes, i.e. all the $g_i = \lceil n_i/r \rceil$ groups should be used. In other words, for packet 1, it should choose a group of r retransmitters; for packet 2 it should choose another group of r retransmitters; and so on for all packets.

In summary, EIDA is a useful mechanism that improves the efficiency and reliability of communication and routing. By reducing congestion, improving routing efficiency, and allocating resources more equitably among nodes, EIDA can lead to a more efficient and reliable network that can handle increased traffic and reduce the risk of communication failures.

4.2 Network division in zones using SLR

SLR [13] is a spatial addressing and routing protocol that comprises two phases: initialization and routing.

The goal of the initialization phase is to assign coordinates to nodes. These coordinates are defined as an integer number of hops from the node to some special nodes called anchors. During this phase, two anchors placed at the vertexes of 2D network broadcast a beacon containing one field representing the current number of hops (similar to a TTL, time to live, field). This field is initialized to zero and increments with each retransmission. At the end of this phase, each zone will have unique coordinates that represent the distance in hops to each of the anchors. Note that all nodes within the same zone have the same coordinates (zone coordinates) and that all the zones are disjoint (i.e. each node belongs to one zone only).

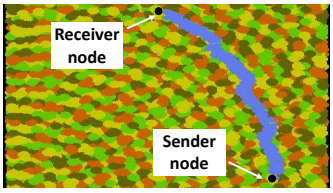

Figure 1: VisualTracer sketch for SLR routing phase.

The goal of the routing phase is to route the data packets from source to destination in a linear routing path based on the coordinates assigned previously, as shown in Fig. 1. In this phase, packets contain the SLR coordinates of the source (sender) and destination (receiver), and each node receiving a packet checks whether it is on the path using a simple formula involving source and destination coordinates; if so, the node forwards the packet, elsewhere it discards it.

4.3 Zone density computation

To run properly, each node in EIDA needs to know the density of its zone. To this end, we implemented a zone density estimator during the SLR initialization phase. We remind that in this phase, each node in every zone sends one SLR packet, so we take advantage of that and append the sender coordinates to it. Afterward, each time a node receives a packet in this phase, it compares the coordinates of the packet sender with its own coordinates: If they are the same, it increments the density of its zone (sender), as shown in Algorithm 2.

Algorithm 2 Zone density computation

```
ZoneDensity ← 0
while node receives a packet do
    if packet_type==SLR_setup then
        if sender_XY==receiver_XY then
            ZoneDensity++
        end if
    end if
end while
```

In this algorithm, *sender_XY* and *receiver_XY* denote the SLR_X and SLR_Y of sender and receiver respectively. *packet_type* denotes the type of received packet (*SLR_setup* in our case), which allows to differentiate it from other packets in the network.

4.4 EIDA algorithm in multi-hop network

EIDA algorithm (Algorithm 1) is executed in each zone separately and independently. The algorithm is updated as follows:

Table 1: Simulation parameters.

Size of simulated area	6 mm * 6 mm
Number of nodes	20 000
Communication radius	350 µm
Packet size	100 bit

(1) packets contain a third field, `slr_xy`, which denotes the sender `slr_x` and `slr_y`, computed as shown in section 4.2: send_packet (`EIDA`, ids, `slr_xy`)
(2) n denotes the density of the current zone instead of the network density. This value is computed during the SLR initialisation phase, as explained in section 4.3.
(3) upon packet reception, an additional check is performed on the equality between the `slr_xy` value in the packet (sender) and of the current node.
if *packet_type==EIDA* AND `slr_xy == my_xy`

To conclude, EIDA mechanism can be applied to multi-zone nanonetworks with slight updates, allowing it to run in each zone separately and independently.

5 EVALUATION

This section evaluates EIDA, in particular the termination and correctness of the algorithm, and compares it with two other assignments using several metrics.

As real experiments are not possible with such a dense nanonetwork, we evaluate our assignment mechanism through simulations. Several nanonetwork simulators exist, but only one, BitSimulator [3], is scalable and allows to simulate dense networks [11]. BitSimulator can simulate tens of thousands of nodes, especially at routing and transport levels. It comes with a visualization program that graphically displays the simulation event. It is free software and has been used to validate the results of several papers[1]. We use it to evaluate our EIDA scheme.

We implemented in BitSimulator EIDA and zone density computation described in Sec. 4.3. All the other protocols were already implemented. We provide a web page[2] to reproduce all the simulation results.

The simulation parameters are shown in Table 1. The nodes are randomly placed in the 2D network using a uniform distribution. We use standard values for TS-OOK modulation: the duration of one pulse (bit) is T_p = 100 fs (cf. "very short symbol duration T_p (i.e., ≈ 100 fs)" [4]), and the time spreading ratio $\beta = T_s/T_p = 1000$ (cf. "The ratio between the time between pulses and the pulse duration is kept constant" [4]).

5.1 EIDA results

In the simulation, we use r=5 (the expected number of nodes in each group) and m=2 (the guaranteed minimum number of nodes in each group). A nanonetwork is multi-hop, hence the algorithm is executed by each zone separately and independently. To show the simulation result, we choose a representative zone, of coordinates (31,25). The density of this zone is n_i=31, as calculated by Algorithm 2. Given these inputs, g = 6 groups, and max_pkts = 12 (cf. Algorithm 1).

[1]http://eugen.dedu.free.fr/bitsimulator
[2]http://eugen.dedu.free.fr/bitsimulator/nanocom23

Table 2: Node id assignment simulation results.

Phase 1 Using packet exchange		**Phase 2** Using backoff			
Node backoff order	id	Node backoff order	id	Node backoff order	ids
1	0	**13**	0	**25**	3
2	1	**14**	1	**26**	1
3	2	**15**	5	**27**	0
4	3	**16**	5	**28**	0
5	4	**17**	0	**29**	1
6	5	**18**	0	**30**	0
7	0	**19**	1	**31**	3
8	1	**20**	4		
9	2	**21**	4		
10	3	**22**	1		
11	4	**23**	5		
12	5	**24**	2		
Continue to 13		Continue to 25			

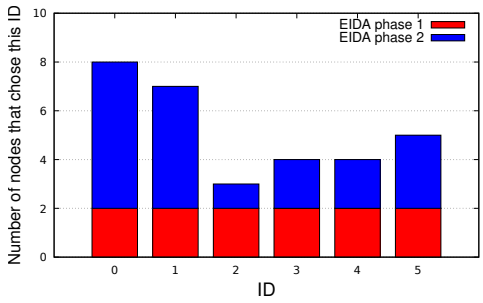

Figure 2: Number of nodes that assign their ids to 0, 1, ..., 5 during packet exchange (phase 1 of EIDA), during backoff (phase 2 of EIDA), and after both phases (sum of both phases).

The results of the assignment are presented in Table 2. In this table, the nodes (1, 2, ..., 32) are ordered from the smallest to the highest backoff. The table confirms that only the first max_pkts = 12 nodes assign their ids in the ideal case (in phase 1, with packet exchange), because they get increasing ids (0, ..., 5). The simulator took 3.5 seconds to finish the id assignment of the 20 000 nodes.

Figure 2 shows the number of nodes that choose each id: using packet exchange (phase 1), using backoff (phase 2), and their sum. In phase 1, the red values show that groups contain an equal number of nodes assigned to them (an equal partition depending on m = 2). In phase 2, the blue values show that without communication, the groups are split into a best effort for equitability, but still non-equal partitions (different numbers of assigned ids to nodes in groups). The sum of these two values shows that EIDA does achieve the guarantee (from phase 1 using packet exchange: there are at least m = 2 nodes in each group in the ideal case of assignment), i.e. it presents the correctness of the algorithm.

Table 3: Number of nodes in each group after id assignment.

	0	1	2	3	4	5
Ideal	6	5	5	5	5	5
EIDA	8	7	3	4	4	5
Random	7	0	5	3	8	8

Table 4: Comparison of ids mechanisms.

	Guarantee ensured	Exchanged packets	Equitability (least squares)
Ideal	yes	$n-1=30$	0.8
EIDA	yes	$nm/r=12$	18.8
Random	no	0	50.8

5.2 Method comparison

The works cited in Sec. 2.3 either assign sparse ids, like the random assignment mechanism (leading to nonequity), or consecutive ids, like the ideal assignment (leading to a high number of packet exchanges). Thus, this section compares the id assignment results of EIDA, random, and ideal assignment mechanisms. The methods are applied on the same zone, which contains 31 nodes.

Table 3 shows the assignment results. The ideal assignment assigns ids sequentially to nodes and incrementally (0, 1, …, 5, 0, 1, …, 5, …), hence, ids are equitably assigned (5 or 6 nodes per group). EIDA results are taken from Fig. 2. Random assignment results are taken from a simple C++ program using the classical Mersenne Twister RNG with `seed=10`[3].

Table 4 compares the three methods in terms of minimum guarantee, number of exchanged packets, and equitability. The equitability uses the least-squares method, calculated by the following equation (where r_i denotes the number of nodes in group i, and $r = 31/6 = 5.1$ their average):

$$S = \sum_{i=0}^{5} (r_i - r)^2 \tag{4}$$

Random assignment does not provide any guarantee: in Table 3 none of the nodes has id =1, which is critical in some applications, such as the routing algorithm presented in Sec. 4.1.

The ideal assignment uses $n-1=30$ packets to assign ids to nodes in this zone, whereas EIDA needs 12 packets (max_pkts), and random assignment needs none.

Looking at the equitability results in Table 3, most groups in the random assignment are far from the ideal assignment, as evidenced by a group with 8 nodes and another one with 0 nodes. In EIDA, each group has at least m=2 nodes, and most of the groups are close to the ideal assignment. The least squares result shows that EIDA ($S = 18.8$) is closer to the ideal assignment ($S = 0.8$) than to the random one ($S = 50.8$).

To conclude, EIDA offers a better trade-off than either the random or the ideal assignment mechanisms alone. It guarantees a minimum number of ids per group while reducing the number of packets exchanged.

[3] `mt19937_64 rng(10); for(i=0;i<31;i++) cout«rng()%6;`

6 CONCLUSION

This paper presents EIDA, a mechanism for assigning ids in the context of dense nanonetworks. EIDA aims to assign best-effort equitable ids to nodes in order to create groups of nodes. It is a combination of ideal and random assignments that depends on the configurable desired guarantee m and the number of nodes r desired in each group.

Compared to the ideal assignment, EIDA uses fewer resources by exchanging fewer packets, while assigning partially equitable ids based on the guarantee value, i.e. not fully equitable as in an ideal assignment case. Compared to random assignment, EIDA requires a specific amount of packet exchange to fulfill a minimum guarantee, making it useful in some applications, contrary to random assignment. Evaluations using a dense nanonetwork simulator illustrate the results and benefits of EIDA.

ACKNOWLEDGMENTS

Carole Al Mawla 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] Carole Al Mawla, Eugen Dedu, Kamal Beydoun, and Dominique Dhoutaut. 2023. FR-SLR, forwarder reduction in SLR routing protocol through zone splitting. In *International Wireless Communications & Mobile Computing Conference (IWCMC)*. Marrakesh, Morocco, 1–6.

[2] Joseph Assaker, Abdallah Makhoul, Julien Bourgeois, and Jacques Demerjian. 2020. A unique identifier assignment method for distributed modular robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Online (Las Vegas, NV, USA), 3304–3311.

[3] Dominique Dhoutaut, Thierry Arrabal, and Eugen Dedu. 2018. BitSimulator, an electromagnetic nanonetworks simulator. In *5th ACM/IEEE International Conference on Nanoscale Computing and Communication (NanoCom)*. ACM/IEEE, Reykjavik, Iceland, 1–6.

[4] Josep Miquel Jornet and Ian F Akyildiz. 2014. Femtosecond-Long Pulse-Based Modulation for Terahertz Band Communication in Nanonetworks. *IEEE Transactions on Communications* 62, 5 (May 2014), 1742–1754.

[5] Filip Lemic, Sergi Abadal, Wouter Tavernier, Pieter Stroobant, Didier Colle, Eduard Alarcón, Johann Marquez-Barja, and Jeroen Famaey. 2021. Survey on Terahertz Nanocommunication and Networking: A Top-Down Perspective. *Journal on Selected Areas in Communications* 39, 6 (Feb 2021), 1506–1543.

[6] Jialiu Lin, Yunhuai Liu, and Lionel M Ni. 2007. SIDA: self-organized ID assignment in wireless sensor networks. In *IEEE International Conference on Mobile Adhoc and Sensor Systems*. IEEE, Pisa, Italy, 1–8.

[7] Yunhuai Liu and Lionel M Ni. 2006. Location-aware ID Assignment in Wireless Sensor Networks. In *International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, Vancouver, Canada, 525–529.

[8] Takuji Nishimura. 2000. Tables of 64-bit Mersenne twisters. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 10, 4 (Oct 2000), 348–357.

[9] ElMoustapha Ould-Ahmed-Vall, Douglas M. Blough, Bonnie S. Heck, and George F. Riley. 2005. Distributed unique global ID assignment for sensor networks. *Ad Hoc Networks* 7, 6 (Nov 2005), 2051–2055.

[10] Junghun Ryu, Jaewook Yu, Eric Noel, and K Wendy Tang. 2011. Node ID assignment in group theoretic graphs for WSNs. In *Wireless Telecommunications Symposium (WTS)*. IEEE, New York, NY, USA, 1–8.

[11] Emre Sahin, Orhan Dagdeviren, and Mustafa Alper Akkas. 2021. An Evaluation of Internet of Nano-Things Simulators. In *6th International Conference on Computer Science and Engineering (UBMK)*. IEEE, Ankara, Turkey, 670–675.

[12] Ye Tian, Min Sheng, and Jiandong Li. 2008. Virtual Grid Spatial Reusing Algorithm for MAC Address Assignment in Wireless Sensor Network. *International Journal of Distributed Sensor Networks* 4, 2 (Apr 2008), 142–154.

[13] Ageliki Tsioliaridou, Christos Liaskos, Eugen Dedu, and Sotiris Ioannidis. 2017. Packet routing in 3D nanonetworks: A lightweight, linear-path scheme. *Nano Communication Networks* 12 (June 2017), 63–71.

[14] Ruiqin Zhao, Yuan Liu, Octavia A Dobre, Haiyan Wang, and Xiaohong Shen. 2020. An efficient topology discovery protocol with node id assignment based on layered model for underwater acoustic networks. *Sensors* 20, 22 (Nov 2020), 6601.