FR-SLR, forwarder reduction in SLR routing protocol through zone splitting

Carole Al Mawla*, Eugen Dedu*, Kamal Beydoun† and Dominique Dhoutaut*
*Université de Franche-Comté, CNRS, institut FEMTO-ST, F-25200 Montbéliard, France
†Faculty of Sciences I, Lebanese University, Beirut, Lebanon
*{carole.almawla, eugen.dedu, dominique.dhoutaut}@univ-fcomte.fr †Kamal.Beydoun@ul.edu.lb

***Abstract*—Current routing protocols are inefficient in dense networks, where nodes have numerous neighbors. An example of a protocol is SLR, which divides the network into zones, with all nodes in a zone sharing the same coordinate. As such, all nodes in a zone participate together in the routing process, leading to the useless consumption of resources. For such networks, we propose FR-SLR, a routing protocol based on SLR, aiming to reduce the number of forwarders in each zone by dividing the nodes into groups and making only one group of nodes forward the packet in each zone. FR-SLR uses an id assignment mechanism to split the nodes in a zone into several groups so that the routing is done per group instead of per zone, thus reducing the number of nodes involved in routing. Simulations on an ultra-dense nanonetwork show the effectiveness of the proposed routing protocol in reducing the number of traffic forwarders.**

***Index Terms*—Dense nanonetworks, routing protocol, forwarder reduction**

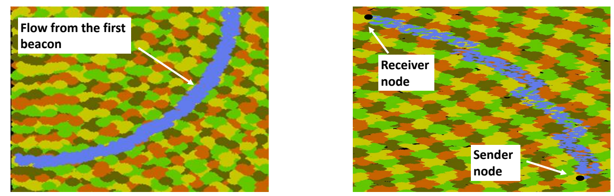
I. Introduction

Nanonetwork consists of a set of interconnected nanomachines (devices a few hundred nanometers or a few micrometers at most in size), which can perform only very simple tasks such as computing, data storage, sensing, and actuation. It enables new applications of nanotechnology in the biomedical field, environmental research, military technology, and industrial and consumer goods applications. A nanonetwork differs from other types of networks by node resources, energy, and memory capabilities that influence the network lifetime, the huge bandwidth, and the ability to multiplex many frames over the same period of time, among others.

Given the tiny size of nodes, nanonetworks are multi-hop and can also have a massive number of nodes, such as tens of thousands, with a high node density (degree), such as hundreds of neighbors per node. The same trend exists in macronetworks, as ultra-dense nanonetworks are emerging as one of the most promising technologies to cope with the tremendously increasing volume of mobile traffic.

In dense nanonetworks, current routing protocols perform poorly, because numerous nodes are involved in the routing process. One such example is Stateless Linear-path Routing (SLR) [1], on which our work is based. It is a spatial addressing and routing protocol. This protocol has two phases: (1) initialization phase and (2) routing phase.

The goal of the initialization phase is to divide the network into zones and assign them coordinates, defined as an integer number of hops from a few special nodes called anchors. In this phase, each of the two anchors placed at the vertexes of 2D network broadcast a beacon, as shown in Fig. 1(a). These packets include a field storing the number of hops from the anchors (it is initialized to zero and increments with each retransmission). At the end of this phase, each zone gets unique coordinates that represent the number of hops to the anchors. It is important to note that *all nodes within the same zone have the same coordinates*.



(a) SLR initialization phase (b) SLR routing phase

Fig. 1. VisualTracer sketch for SLR initialization and routing phases.

The goal of the routing phase is to route the data packets in a linear routing path based on the coordinates assigned in the previous phase, as shown in Fig. 1(b). During routing, each node that receives a packet checks, using a simple formula, whether it is on the path based on its own coordinates and the source and destination coordinates (found in the packet). If so, the data packet is forwarded; elsewhere, it is discarded.

Given that in SLR all the nodes in a zone share the same coordinates, *all the nodes* in the routing zones retransmit the packet. In this paper, we propose a method to reduce the number of nodes during packet routing. It works by dividing the nodes in each zone into different groups, using a node id assignment mechanism, and making only one group reforward the packet instead of all the nodes in each zone. Consequently, this reduces the number of packets forwarded, leading to less resource consumption and energy, and increasing the network lifetime.

The contribution of this paper is to propose FR-SLR, an SLR-based routing protocol efficient in a multi-zone dense network, based on the aforementioned grouping idea. Simulations done on a scalable nanonetwork simulator show a notable reduction in terms of number of forwarders when compared with related forwarding methods.

This paper is organized as follows. The related work on reducing the number of forwarders is presented in section II.

Section III describes the proposed FR-SLR routing protocol using various id assignment mechanisms. Section IV evaluates FR-SLR using one such assignment and compares it to other flooding and destination-oriented protocols. Finally, the conclusion is drawn in section V.

II. Related work

This section presents routing protocols that have the same goal as FR-SLR, which is to deliver the packet from source to destination with a reduced number of forwarders. We group them based on the approach they use.

A. Cluster-based approaches

Cluster-based approaches first divide the network into clusters and afterward route packets using cluster heads.

To create clusters, [2] bundle adjacent small cell base stations into a group. Clustering in this paper is done in two steps, (1) cluster splitting/creation and (2) combining them with a weighted k-means algorithm. The k-means group the small base stations into k clusters according to their distances (location and traffic load of the sub-cluster). [3] proposes a swarm intelligence-based fuzzy routing protocol to generate balanced clusters over the network with the ability to determine the precise number of clusters by considering the residual energy, distance to the sink, and distance from the cluster centroid to select cluster heads.

Routing is done exclusively by cluster heads (CH), and each cluster has a cluster head. This consumes the energy of the CH and creates a load balancing problem. On the contrary, in FR-SLR, a *different* group of nodes is chosen each time in each zone to retransmit the packet. For example, [4] proposes a hierarchical clustering method where the nodes in the cluster can transmit data to the CH in one hop. Then, the CH nodes transfer data to the nanocontroller (NC) through multi-hop, which has a larger volume and stronger calculation processing capability than the nanonode. It solves the common problem of balancing the energy consumption in the network by continuously updating CH, where the node with the highest residual energy is selected as CH. In this paper, the energy of NC is not limited, which is not the same case as in a dense nanonetwork. As another example, [5] introduces a fuzzy logic-based mobility management solution for mobile cluster-based WNSNs to deliver the data packet to static NC. The communication between the nanosensor and the NC of its cluster is done in one hop. The decision to select the NC (stationary nodes) is based on three criteria: the distance of mobile nanonode from NC, residual energy, and the traffic load of this NC. Subsequently, the nanocontroller assigns specific slots for data transmission. Afterward, every nanonode, according to the TDMA schedule, transmits packets to its parent nanocontroller, which then aggregates and forwards them to the nano-micro interface. Yet this protocol cannot be adopted in dense networks because of its computational complexity, the large memory needed, and the need for stationary nodes (NC) distributed within the network.

B. Flooding protocols

Pure flooding is the simplest routing method, where every node in the network forwards once each received packet. This flooding generates a significant amount of messages through the network, resulting in broadcast storms in dense networks.

In probabilistic flooding, nodes forward the packet with a certain probability [6]. The chosen probability can be fixed and can depend on several factors, such as density, distance, and speed. The probability needs to be tuned to prevent broadcast storms and guarantee message delivery.

Backoff flooding is a high-quality flooding that uses a counter to count the copies received and a waiting time called "backoff" for the packet forwarding [7]. The backoff value is selected randomly from the backoff window. This window is very large and is proportional to the neighbor density. A node forwards the packet only if it has not received r (redundancy) copies of the same packet during the backoff time.

C. Destination-oriented protocols

In flooding methods, the packet is flooded into the whole network instead of being transmitted toward the destination, leading to useless packet reception and resource waste (e.g., bandwidth and energy). Destination-oriented protocols, such as SLR, counter-based SLR, and Maze-routing, reduce packet dissemination in unnecessary directions and reduce the number of forwarders.

SLR has been presented in the introduction. Counter-based SLR combines SLR with backoff flooding: A node that receives a packet forwards it only if it has seen fewer copies of the packet than a given threshold.

Maze-routing [8] is a distributed routing algorithm that guarantees delivery or indicates that the destination is unreachable by adding extra fields to a message. It allows routing around fault regions without requiring nanonodes to store any information other than their coordinates or the need for a routing table (close to the idea of SLR). Yet, the route may be far from optimal, whereas FR-SLR uses the optimal path (much shorter in some cases).

D. Sleeping-based approaches

Another way to reduce the number of forwarders is to use a sleeping mechanism, also known as duty cycling, which turns sensor nodes on and off when necessary. In such a mechanism, a node always wakes up long enough to allow the reception of one or more frames. Afterward, it goes back to sleep by turning off most of its reception and processing capabilities. For example, in cluster synchronization [9], sensor nodes are grouped into synchronized clusters. In the same cluster, sensor nodes wake up or go to sleep at the same time. But clusters act together with others asynchronously. In [9], all the decisions are made by the controller, which is usually connected to a constant power supply, which is not appropriate for dense networks.

Furthermore, the sleeping mechanism can be decentralized; each node has its own wake-up and sleep schedule. In [10], the time axis is divided into a number of time slots, and each node autonomously decides to sleep, listen, or transmit in a time slot. The decision is based on its current situation and an approximation of its neighbors' situations without the need for communication with neighbors.

SERENA [11] is a decentralized node activity scheduling algorithm based on three-hop coloring. It allows nodes to sleep while ensuring end-to-end communication by assigning time slots to nodes. Slots are assigned to nodes based on their color. Any node stays awake only during its slots to transmit and the slots assigned to its 1-hop neighbors to receive their messages; it sleeps the remaining time. Two nodes that are 1, 2, or 3-hop neighbors have different colors. Hence, color is reused four hops away.

FR-SLR is a destination-oriented protocol, which is why, for a fair comparison, the proposed protocol is compared to destination-oriented protocols (SLR and counter-based SLR) and not only to flooding (pure, backoff, and probabilistic) protocols. This comparison takes place in section IV-B and uses the same scenario.

III. Forwarder reduction in SLR routing (FR-SLR)

We recall that SLR has a big drawback: during routing, *all* the nodes in the zones on the path from source to destination retransmit the packet, as shown in Fig. 1(b). In ultra-dense networks, this consumes a lot of energy and reduces network lifetime. FR-SLR aims to avoid this resource waste and is described in the following.

A. FR-SLR description

In FR-SLR, nodes from each zone are divided into g groups; all the nodes in a group have the same id and groups have different ids. Each packet has an additional field specifying an id initialized by the source of the packet with its own id. In each SLR zone on the routing path, only the nodes whose id is the same as the id field in the packet retransmit the packet (instead of all of them like in SLR).

Node ids are selected using any id assignment mechanism, provided that the following condition is met: ids start from 0 and follow in sequence, without holes, i.e. 0, 1, 2, ..., g, where g is chosen by the user and given as input in the assignment mechanism. g value is known by all the nodes. We will present a few such mechanisms in the next section.

Users can select g based on multiple factors, such as the density of zones, zone size, and application. If the user needs, for example, 5 forwarders in each zone and the zone density is 10 nodes, the user must set $g = 2$, whereas if zone density is 100, the user must set $g = 20$. Additionally, for example, if the zone space is small and all the nodes are within the communication range of each other, a small g can be chosen, whereas if zones are large, then g needs to be bigger to prevent die out. Also, critical applications might choose a small value for g in order to have more nodes retransmit the packet.

We note that, as it is, the method leads to a load balancing problem in some cases. For example, in a scenario where a source node sends 1000 packets, all the packets contain the same id, hence *the same group* in each zone of the path reforward the packet. To avoid this case, FR-SLR nodes also use the packet sequence number to decide the group (provided that packets have such a field), so that packets from the same source are processed by *different* groups.

The full algorithm is shown in Algorithm 1. is_on_path is the SLR function that determines whether the current node is on the path between the source and destination based on their coordinates.

FR-SLR differs from SLR by adding an additional test (condition). This test compares the id equality of the packet sender to the id of the current host. Thus, the current host only forwards the packet if this condition is met, i.e., if they are holding the same id, which means that they belong to the same group.

Algorithm 1 FR-SLR pseudocode

Upon packet pkt reception in node_i:
$coord\_i \leftarrow get\_coord(node\_i)$
$coord\_src \leftarrow get\_source\_coord(pkt)$
$coord\_dst \leftarrow get\_destination\_coord(pkt)$
if is_on_path ($coord\_i$, $coord\_src$, $coord\_dst$) **then**
  $src\_id \leftarrow get\_source\_id(pkt)$
  $id \leftarrow get\_id(node\_i)$
  $seq\_no \leftarrow get\_sequence\_nb(pkt)$
  if $(src\_id + seq\_no) \bmod g$ **==** id **then**
    forward_packet (pkt)
  end if
end if

To conclude, in FR-SLR, only a part of the nodes in each zone on the path retransmit the packet based on node id.

B. Integration of FR-SLR with existing id assignment mechanisms

FR-SLR uses node ids to divide the forwarders (reduce the number of forwarders) of each packet. In this section, we present various mechanisms to assign each node an id.

1) Random id assignment: In the random id assignment, each node assigns itself a random id from 0 to $g-1$. An example of random assignment is shown in Table I. Its results are taken from a simple C++ program using the classical Mersenne Twister RNG with `seed=10`[1]. This table also shows that the number of nodes in the group of id=1 is zero. Thus, when a packet is sent by a sender with $src\_id + seq\_no = 1$ and reaches a zone where group 1 is empty, the packet is not forwarded, leading to a die-out. We will therefore not use the random assignment in the evaluation.

2) Ideal id assignment: The ideal id assignment divides the nodes into g groups, each with the same ($\approx$) number of nodes, $\left[\frac{n}{g}\right]$ or $\left[\frac{n}{g}\right]+1$ nodes. Thus, the partition is $\approx 100\%$ equitable, as shown in Table I. However, to obtain this equal partition, the number of packet exchanges in each zone is $n-1$, a big

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

TABLE I
NUMBER OF NODES IN EACH GROUP AFTER ASSIGNMENT.

	0	1	2	3	4	5
Ideal	6	5	5	5	5	5
EIDA	9	4	5	4	5	4
Random	7	0	5	3	8	8

TABLE II
SIMULATION PARAMETERS.

Parameter	Value
Size of simulated area	6 mm * 6 mm
Number of nodes	20 000
Communication radius	285 µm
Packet size	100 bit

number in ultra-dense networks, hence we will not use this assignment in our evaluation.

3) EIDA assignment mechanism: EIDA partitions several nodes into groups of the same id [12]. It assumes that nodes can communicate with each other. Thus, multi-hop networks need to be partitioned into zones first, and afterward, EIDA can be used in each zone separately and independently. SLR does such a partitioning in the initialization phase, and EIDA needs to start after the SLR initialization phase.

EIDA provides two configurable parameters, r and m, where r is the expected number of nodes in each group and m the minimum number of nodes in each group (with $m \leq r$). EIDA assigns each node to one of the g groups (0, 1, 2, ..., $g-1$) in a best-effort equitable manner, respecting the condition on m. The number of groups g is computed as $g = \left\lceil \frac{n}{r} \right\rceil$ (ceiling operation), where n denotes the density of the current zone. EIDA results are shown in Table I.

During the first phase of EIDA, nodes start by communicating in order to ensure that m nodes have been assigned to each group. In the second phase, the remaining nodes do a random assignment without any communication. This reduces the number of packet exchanges.

EIDA has a useful property in our case. Given that any group has at least m nodes (as shown in Table I), during routing each packet is forwarded by at least m nodes in each zone, thus preventing the die-out problem. We will therefore use EIDA to assign node ids for FR-SLR evaluation.

It should be noted that EIDA has a limitation affecting FR-SLR: when combined with EIDA, FR-SLR does not work in *heterogeneous* networks. The reason is that, contrary to the other assignment methods, EIDA does not use a fixed g value as input but instead computes g based on m and r. In heterogeneous networks, where zones have different densities, the g value can be different, otherwise said, some zones can have e.g. 5 groups (50 nodes divided by r=5), others 2 groups (20 nodes divided by r=5). Thus, in the case where the source is in group 3, the packet will have id 3 and when crossing zones with only 2 groups, no node exists with group 3, hence the packet is not retransmitted and a die-out occurs.

IV. EVALUATION

This section evaluates the FR-SLR routing protocol with the EIDA mechanism and compares it to other routing protocols. Furthermore, the motivation for this paper is revealed when the simulation compares the number of packet exchanges using SLR and FR-SLR.

An example of dense networks is nanonetworks. They currently use the TS-OOK (Time Spread On-Off Keying) modulation [13], which is based on femtosecond-long pulses in the terahertz band, appropriate to the very limited energy of nanonodes. Bits are sent using a sequence of pulses interleaved by a randomly selected constant duration.

As real experiments are not possible with such a dense network, we evaluate our routing protocol through simulations. Implementation and evaluation of the FR-SLR algorithm are done using BitSimulator [14], the only simulator allowing simulation of ultra-dense nanonetworks (tens of thousands of nodes). It comes with a visualization program that displays graphically the simulation events. It is free software and has been used to validate the results of several papers[2].

We implemented FR-SLR forwarding in BitSimulator. The other routing protocols were already included in the simulator.

The simulation parameters are shown in Table II. The nodes are placed randomly in the 2D network, using a uniform distribution, and are static. We use standard low-level parameters for TS-OOK modulation, i.e. duration of one pulse (bit) T_p = 100 fs [13]), 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" [13]).

We provide a web page[3] to reproduce all the simulation results.

A. Die-out avoidance in FR-SLR with EIDA

This section shows that FR-SLR/EIDA avoids the die-out, i.e. packets reach the destination.

As a node id assignment mechanism, we use EIDA because it uses fewer packet exchanges than the ideal assignment, and the random assignment can lead to die-out, as shown in Sec. III-B. EIDA parameters are r=5 (expected number in each group) and m=2 (minimum number of nodes in each group). An example of an assignment is presented in the previous Table I, for the zone of coordinates (31,25), shown in Fig. 2, with 31 nodes. The table confirms that EIDA guarantee of m is achieved because each group contains at least $m = 2$ nodes.

In the scenario, we chose a sender with coordinates (35,25) and a receiver with coordinates (16,29). The simulation uses the first packet, hence $seq\_no = 0$.

In the FR-SLR simulation, Table III presents the number of forwarders in each zone that belong to the transmission path between the sender (35,25) and receiver (16,29). This table shows that for each zone, the number of forwarders is greater than the guarantee ($m = 2$). Thus, FR-SLR inherits from EIDA the guarantee of minimum group size, so no die-out can occur.

[2]http://eugen.dedu.free.fr/bitsimulator
[3]http://eugen.dedu.free.fr/bitsimulator/iwcmc23

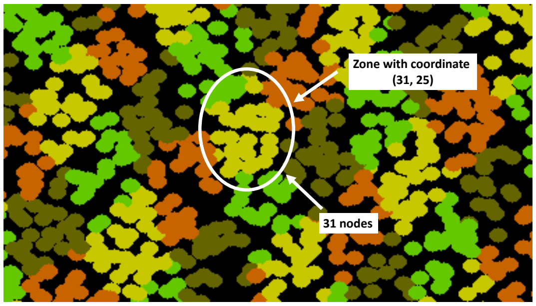


Fig. 2. VisualTracer sketch for the chosen zone, of coordinates (31,25).

TABLE III
NUMBER OF FORWARDERS IN EACH ZONE IN FR-SLR.

Zone	SLR coordinates	Number of forwarders
0	35, 26	6
1	34, 26	10
2	33, 26	6
3	32, 26	2
4	31, 26	4
5	31, 27	7
6	30, 27	4
7	29, 27	3
8	28, 27	6
9	27, 27	4
10	26, 27	4
11	26, 28	5
12	25, 28	4
13	24, 28	3
14	23, 28	5
15	22, 28	2
15	21, 28	4
17	21, 29	6
18	20, 29	9
19	19, 29	6
20	18, 29	5
21	17, 29	5
22	16, 29	5

To conclude, FR-SLR with EIDA avoids the die-out.

B. Comparison of FR-SLR/EIDA to other routing protocols

In this section, we compare FR-SLR/EIDA to five routing protocols: SLR, counter-based SLR, pure flooding, backoff flooding, and probabilistic flooding. The comparison is based on the number of forwarders using the aforementioned scenario.

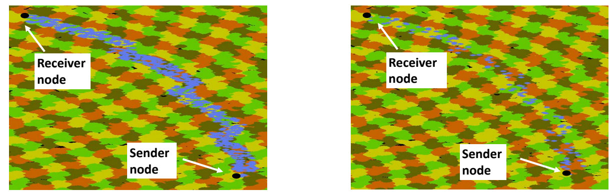
Counter-based SLR uses a redundancy of r=5, similar to FR-SLR (for a fair comparison).

Table IV presents the number of forwarders in each routing protocol. In SLR, there are 619 forwarders, because *all the nodes* in a zone on the path reforward the packet. In FR-SLR, since only *one group* of nodes in each zone retransmits the packet, there are 116 forwarders, as can be computed by summing up the numbers in the last column in Table III (+1 for the first packet). The forwarders in both cases are presented as blue points in Fig. 3. Counter-based SLR has a comparable number of forwarders (a bit higher in this case).

In a zone with a density of around 30 nodes and a parameter of $r = 5$ for FR-SLR, there are $\left\lceil \frac{30}{5} \right\rceil = 6$ groups in this zone.

TABLE IV
COMPARISON OF VARIOUS ROUTING PROTOCOLS.

Method	Number of forwarders	Simulation time (ns)
Flooding methods:		
Pure flooding	20 004	158
Probabilistic flooding	718	277
Backoff flooding	575	1544
Destination-oriented methods:		
SLR	619	168
Counter-based SLR	147	5361
FR-SLR	116	198



(a) SLR routing (b) FR-SLR routing

Fig. 3. VisualTracer sketch for SLR and FR-SLR routing phase.

Therefore, $\frac{1}{6}$ of the nodes (one group) in each zone reforward the packet. If the linear path between the sender and receiver contains 23 zones, the total number of forwarded packets in FR-SLR should be reduced by a factor of 6 compared to SLR. This is proved in Table IV, where the number of forwarded packets of FR-SLR is approximately equal to $\frac{1}{6}$ of the number of forwarded packets of SLR.

Table IV shows the time to deliver the packet from source to destination (the delay). FR-SLR is much faster than counter-based SLR and backoff flooding (because these protocols need a large backoff during each forwarding). The difference in time, of thousands of ns, i.e. 1 µs, seems small because it is at a small scale of 23 zones between source and destination, but on longer paths, the difference becomes bigger.

C. Discussion on FR-SLR vs counter-based SLR

Given that counter-based SLR and our FR-SLR method have a comparable number of forwarders, in this section we compare them more thoroughly. Table V compares both protocols analytically.

In FR-SLR, if all the nodes that are chosen to retransmit the packet (based on the group id) fail, a die-out occurs. On the contrary, in counter-based SLR, no die-out occurs since nodes are chosen randomly each time. We plan to improve on this in future work.

TABLE V
FR-SLR VS COUNTER-BASED SLR.

Property	FR-SLR	Counter-based SLR
Works in node failures	no	yes
Initialization phase	yes	no
Backoff during routing	no	yes
Redundancy	per zone	per communication range

FR-SLR, given that it uses the id of nodes, requires an id initialization mechanism before the routing protocol starts, as opposed to counter-based SLR, where no initialization phase is needed.

For each packet forwarding, the counter-based SLR protocol uses a large waiting time (backoff), whereas in FR-SLR there is no backoff since nodes simply compare the id of the packet sender and the current host (itself) to decide whether to forward the packet or not.

In the counter-based SLR protocol, redundancy applies to the communication range, which can contain one or multiple zones, whereas in FR-SLR redundancy applies per zone, i.e., in each zone r nodes reforward the packet.

The redundancy has many consequences on die-out. Some die-out problems appear in counter-based SLR, depending on the communication ranges used in the initialization and in routing SLR phases. The same communication range (220) for both phases results in a die-out problem approximately in the middle of the transmission path. The cause is that the redundancy is used in the communication range area, not the zone area as in FR-SLR. Hence, when the communication range of the nodes responsible for transmitting the packet does not propagate to the next zone in the transmission path, a die-out occurs. The same problem appears if the difference between the two communication ranges is small; for example, using r=1 and (220, 250) for the (initialization, routing) communication ranges, a die-out occurs again since the communication range of the node responsible for retransmission is too small to reach the next zone.

To conclude, the proposed routing protocol FR-SLR uses generally fewer forwarder packets and smaller time to deliver packets compared to the mentioned flooding (pure flooding, probabilistic flooding, backoff flooding) and destination-oriented protocols (SLR, counter-based SLR), albeit the communication ranges of nodes play an important role in this comparison.

V. Conclusion

This paper introduces FR-SLR, an enhanced SLR routing protocol, using the EIDA mechanism (a combination of ideal and RNG assignments) in the context of dense networks. It reduces the number of forwarders during packet routing by grouping nodes in each zone. Grouping is done using the id assignment method.

Evaluations are done using a dense nanonetwork simulator and illustrate the benefits of FR-SLR when combined with EIDA as an assignment mechanism. FR-SLR outperforms the compared protocols, either flooding or destination-based ones, in terms of the number of forwarders and time to deliver packets.

Future work includes improving FR-SLR to make it avoid transmission die-out in the case of node failures.

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] A. Tsioliaridou, C. Liaskos, E. Dedu, and S. Ioannidis, "Packet routing in 3D nanonetworks: A lightweight, linear-path scheme," *Nano Communication Networks*, vol. 12, pp. 63–71, Jun 2017.

[2] J.-W. Park, E.-H. Kim, and E.-K. Hong, "Cluster splitting and combining in ultra-dense networks," *IET Communications*, vol. 14, no. 9, pp. 1438–1443, Apr. 2020.

[3] Z. M. Zahedi, R. Akbari, M. Shokouhifar, F. Safaei, and A. Jalali, "Swarm intelligence based fuzzy routing protocol for clustered wireless sensor networks," *Expert Systems with Applications*, vol. 55, pp. 313–328, Aug 2016.

[4] J. Xu, Y. Zhang, J. Jiang, and J. Kan, "An energy balance clustering routing protocol for intra-body wireless nanosensor networks," *Sensors*, vol. 19, no. 22, p. 4875, Nov 2019.

[5] N. Rikhtegar, R. Javidan, and M. Keshtgari, "Mobility management in wireless nano-sensor networks using fuzzy logic," *Journal of Intelligent & Fuzzy Systems*, vol. 32, no. 1, pp. 969–978, Jan 2017.

[6] D. G. Reina, S. Toral, P. Johnson, and F. Barrero, "A survey on probabilistic broadcast schemes for wireless ad hoc networks," *Ad Hoc Networks*, vol. 25, pp. 263–292, 2015.

[7] T. Arrabal, D. Dhoutaut, and E. Dedu, "Efficient density estimation algorithm for ultra dense wireless networks," in *27th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2018, pp. 1–9.

[8] M. Fattah, A. Airola, R. Ausavarungnirun, N. Mirzaei, P. Liljeberg, J. Plosila, S. Mohammadi, T. Pahikkala, O. Mutlu, and H. Tenhunen, "A low-overhead, fully-distributed, guaranteed-delivery routing algorithm for faulty network-on-chips," in *9th International Symposium on Networks-on-Chip*, New York, NY, USA, Sep 2015, pp. 1–8.

[9] Y. Wang, H. Chen, X. Wu, and L. Shu, "An energy-efficient SDN-based sleep scheduling algorithm for WSNs," *Journal of Network and Computer Applications*, vol. 59, pp. 39–45, May 2016.

[10] D. Ye and M. Zhang, "A self-adaptive sleep/wake-up scheduling approach for wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 48, no. 3, pp. 979–992, Mar. 2017.

[11] P. Minet, S. Mahfoudh, G. Chalhoub, and A. Guitton, "Node coloring in a wireless sensor network with unidirectional links and topology changes," in *IEEE Wireless Communication and Networking Conference*. Sydney, Australia: IEEE, Apr 2010, pp. 1–6.

[12] C. Al Mawla, E. Dedu, K. Beydoun, and D. Dhoutaut, "EIDA, a best effort equitable distributed id assignment mechanism for heterogeneous dense network," *Submitted to the 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–7, 2023.

[13] J. M. Jornet and I. F. Akyildiz, "Femtosecond-long pulse-based modulation for terahertz band communication in nanonetworks," *IEEE Transactions on Communications*, vol. 62, no. 5, pp. 1742–1754, May 2014.

[14] D. Dhoutaut, T. Arrabal, and E. Dedu, "BitSimulator, an electromagnetic nanonetworks simulator," in *5th ACM/IEEE International Conference on Nanoscale Computing and Communication (NanoCom)*. Reykjavik, Iceland: ACM/IEEE, Sep. 2018, pp. 1–6.