Ring-based forwarder selection to improve packet delivery in ultra-dense networks

Farah Hoteit*, Eugen Dedu*, Winston K.G. Seah† and Dominique Dhoutaut*
*FEMTO-ST Institute, Univ. Bourgogne Franche-Comté, CNRS
Montbéliard, France
Email: firstName.lastName@univ-fcomte.fr
†School of Engineering and Computer Science, Victoria University of Wellington
Wellington, New Zealand
Email: winston.seah@ecs.vuw.ac.nz

***Abstract*—This paper proposes a novel protocol-agnostic approach to optimize the performance of routing protocols in *ultra-dense networks* through a careful selection of the forwarders in a multi-hop transmission. In an ultra-dense wireless network, nodes have hundreds of neighbours, and existing routing protocols which require neighbourhood information are unable to operate efficiently. Using our method, each node wanting to transmit first selects forwarders that fall in a *ring* near the border of the communication range of the transmitting node, which makes up a subset of all the node's neighbours. This significantly reduces the number of nodes contending for the wireless channel yet ensures that there are sufficient forwarders to deliver packets successfully. We validate our approach using two routing schemes, one flooding and one unicast, augmented with our forwarder selection method and applied to an electromagnetic nanonetwork scenario as a novel incarnation of ultra-dense networks. Simulations using an enhanced propagation model show that our forwarder selection method drastically reduces the number of forwarders while still allowing packets to reach the intended destinations.**

***Index Terms*—Ultra-dense network, scalability, forwarding, congestion, nanonetwork.**

I. Introduction

Wireless ad hoc networks, i.e. infrastructure-less radio networks, are nowadays ubiquitous. In these networks, *multi-hop* communications allow to reach a destination beyond the range of a given node with the help of other nodes acting as routers (relays or forwarders) [1], and the spatial reuse they provide enables higher global and application throughput [2].

While many networks have a large number of nodes, more and more networks are experiencing an increase in the number of neighbours per node. The number of neighbours of a node is usually known as *node degree*, and also neighbour density, neighbourhood density, or local density, all of which are related to the network density. With the proliferation of connected objects in the Internet of Things (IoT), and their increasing communication ranges (low power wide area networks, LPWAN), the number of neighbours in the communication range of a node is increasing, and networks are becoming denser.

Current wireless routing protocols are usually not designed for ultra-dense networks. By *ultra-dense* we do not only mean a large number of nodes, but a number of neighbours so high (hundreds or thousands of 1-hop neighbors) that most classical protocols fail or become inefficient (this should not be mixed up with ultra-dense 5G networks (UDN) for example, which concern the ratio between access points and users). The inefficiency of classical protocols appears for all protocols which ignore the network density. For example, a pure flooding in a network of density 100 nodes quickly leads to the well-known *broadcast storm problem* [3]: the 100 neighbours that received the initial packet try in turn to send it again in a very short time and space, causing contention on the channel, collisions, waste of energy and packet losses. A simple solution would be to reduce the proportion of forwarders in the network. However, one has to be careful to not decrease it too much, as the reverse problem can manifest itself. If not enough nodes participate in the forwarding of packets, the *die out* problem can appear, where the propagation stops without reaching all the nodes.

The inefficiency of packet propagation has been solved in the past literature, but the proposed methods have various constraints unsuitable in ultra-dense networks: they either use coordinates of all neighbours to compute which of them is nearer the destination, or use heavy storage or computations, or need hardware modules such as geo-positioning system (e.g. GPS, Global Positioning System) or RSSI (Received Signal Strength Indicator).

In this paper we propose a simple scheme to select a *ring* of forwarders which solves the problem of too many forwarders and does not have aforementioned constraints. The only assumption we use is that nodes can send packets with different transmission powers. It basically works like this: each node, prior to sending its *first data packet*, sends two control packets at different powers; afterwards all the transmissions occur normally. Nodes near the sender receive both control packets, whereas nodes which have received only one of the two control packets lie on a ring near the border of the communication range. *Only the nodes in the ring forward the subsequent data packets from that node*. Consequently, our scheme *selects* geographically the nodes acting as forwarders, which implies reducing the number of forwarders.

This ring-based forwarder selection method is not meant to be a standalone mechanism, but rather integrated into routing or data delivery protocols. It can be used to implement an optimized global broadcast (i.e. flooding data in the whole network), but can also optimize other unicast or multicast protocols that rely on a local broadcast for their basic operations.

We implemented the ring in a simulator of electromagnetic nanonetworks, consisting of nanodevices [4]. Nanonetworks are envisioned to revolutionise some fields [5]: programmable metamaterials to mitigate wireless channel losses [6], healthcare applications for vital signs monitoring, detection of cardiovascular abnormalities, various applications enabled by the Internet of NanoThings [7], and can perform tasks of sensing and actuation at an unprecedented small scale. The choice of a nanonetwork is to bring further a recent incarnation of ultra-dense networks with resource-constrained devices, where traditional routing protocols cannot be applied. However, despite choosing nanonetworks to validate our method, the ring principle stays the same in any other ultra-dense wireless multi-hop network.

We present the results of an extensive evaluation. We test our scheme on two different scenarios and two different higher layer protocols (an optimized form of flooding and one with a specific destination). We analyse the effect of the ring on the number of forwarders, the number of receivers (packet delivery) and the data traffic. The results show that, for both protocols, the ring is highly efficient compared to without ring.

The contribution of this paper is twofold. First, we propose a simple method to create a ring where the potential forwarders are found. The scheme works in any ultra-dense wireless network, provided that nodes can send packets using different transmission powers. It can be integrated into higher level broadcasting or routing protocols. Second, we conduct an extensive simulation study to analyse how different features (such as control packet transmission power, location of nodes, and knowledge of neighbours) affect the performance.

The next sections present the related work, followed by the details of the ring principle. We then evaluate the ring using extensive simulations, before concluding the paper.

II. Related work

Our proposed protocol is based on a greedy strategy, which aims essentially to minimize the overlapped broadcast in networks, but also shares similarities with certain aspects of mobile ad hoc routing protocols, discussed in the following.

A. Optimized Link State Routing Protocol

Optimized Link State Routing Protocol (OLSR) is a proactive, table-driven routing protocol for mobile ad hoc networks [8]. OLSR selects a minimum set of neighbouring nodes called multipoint relays (MPRs) such that they cover all the 2-hop neighbours. OLSR incurs a high overhead in ultra-dense networks, because nodes need to know their 2-hop neighbours. It also has to compute the optimal subset of 1-hop neighbours, which is an NP-complete problem, or use sub-optimal (but more efficient) heuristics. MPRs selection can still be used in denser mesh networks (up to 150 nodes) [9], but requires powerful hardware for mesh nodes and a sufficiently stable network.

B. Greedy forwarding strategies and protocols

We can cite several protocols: in Geometric broadcast in dense wireless sensor networks [10], a node picks a set of neighbours as forwarding candidate nodes located at the boundary of the communication range based on a virtual hexagon-based coverage, without needing a GPS to locate them. The assumption is that each node knows its 1-hop and 2-hop neighbour IDs. The sender calculates the set of common nodes with his 1-hop neighbour to estimate the location of its neighbour. On the contrary, our strategy is receiver-initiated where the neighbour decides to forward or not (more scalable) and it does not require nodes to *locate* their neighbours.

In Contention-based forwarding for mobile ad hoc networks [11] and Blind geographic routing for Sensor Networks (BGR) [12], nodes set timers and the node whose timer expires first is assumed to be closer to the destination and thus forwards, whereas the other nodes cancel their transmission. This greedy approach cannot be applied in our context as the location or direction of destination node is unknown to nodes. In edge forwarding [13], nodes keep track of all their neighbours and require the farthest ones to be forwarders. In Greedy Perimeter Stateless Routing (GPSR) [14], a node knows its location and the locations of its neighbours, and packets contain the coordinates of the destination. The neighbour closest to the destination is selected as forwarder. It uses perimeter routing to route around dead ends (mathematically avoiding local minimums).

In vehicular ad hoc networks (VANETs), where GPS is usually a given, greedy forwarding [15], [16], [17] selects the farthest vehicle from the transmitter as the new forwarder.

All these protocols however have limitations, either in their scalability (when networks are ultra-dense) or in their applicability (when nodes are very simple). Scalability is a problem especially for protocols that need an extended knowledge of their neighbourhood (especially if they require knowledge beyond 1-hop). They need a lot of messages and memory to maintain a correct view of their environment. Applicability mainly includes assumptions on the available hardware and resources. GPS, which is a given in VANETs, is not feasible in potentially resource-constraint networks. Received Signal Strength Indication (RSSI), which is often used as an alternative distance measuring technique, may not be available either in very simple transceivers. Memory and computational power may also be heavily restricted.

Finally, like our method, some aforementioned works select forwarders at the border of communication range. Compared to them, our method differs in the constraints, as it considers ultra-dense networks (where nodes can have thousands of neighbours), where neither location information nor signal strength measurement are available. Moreover, our method targets ultra-dense networks and can effectively work even if the nodes have constrained resources.

III. Ring scheme

In a wireless network, when a node sends a packet, all the nodes in its communication range (called *neighbours*) receive it. In a multi-hop transmission, the forwarding nodes are chosen among these neighbours. Depending on the routing algorithm, the forwarding nodes can be all of them (e.g. in pure flooding), or only some of them (e.g. in backoff flooding). In both cases, an inefficiency appears regarding the position of forwarders. For routing purposes, instead of choosing the forwarders randomly, it is more efficient to choose them among furthest neighbours (near the border of communication range), because they lead to fewer hops (or packets generated) towards destination. Our scheme considers this fact, and makes only the neighbouring nodes close to the border to forward it.

Note that the location information is unavailable to nodes, and thus sectoring for example cannot be performed in flooding schemes. Moreover, the neighbourhood and network knowledge is also unavailable so even in a grid-like network where nodes know their own coordinates, they still do not know their neighbours coordinates. Consequently, a node cannot rely on coordinate information. On the contrary, the only key assumption used by the ring protocol is that nodes can change the power used to transmit packets.

Our scheme works as following. Before forwarding *the first data packet*, a node sends two control packets with different powers: one at (or close to) maximum power and the next at a lower power. The nodes having received only the first packet and not the second one form a ring at the edge of the communication radius (the difference of two concentric discs). Nodes memorize the transmitter ID (no matter the initial source of the packet) and a boolean value as whether it is on the transmitter's ring or not. Only the nodes within the ring are selected as potential forwarders. (Note that the list of neighbors a node is in the ring of is much smaller than the list of all the neighbors. Also, in a network with mobile nodes, the control packets could be sent several times, e.g. each time the network changes, or at regular predefined intervals.)

Depending on the routing algorithm, all the ring nodes (in e.g. pure flooding) or only some of them (e.g. in backoff flooding) will *effectively* forward the packet. The forwarding algorithm is formally presented in Algo. 1. Note that the network is dense enough to always have nodes on the ring with an appropriate ring width.

It is worth to note that in the simple unit disc model of propagation (all or nothing, i.e. a packet is received by a node if and only if it is inside the communication range of the sender), the communication range is a circle which results in a perfect ring. Instead, in the more realistic shadowing model, mimicking an imperfect transmission range, the packet loss probability increase with the distance from the source. As such, some nodes may receive the low-power packet, and *not* receive the high-power packet. Our method works in this case too, and the evaluation part uses the shadowing model.

Algorithm 1: Ring algorithm executed by each node.

```
Initially: table = empty, ctrlSent = false
Upon packet reception p:
n = previous node (sender at "MAC"-level) of the packet
if p is a high-power packet then
    table[n,0] = true
else if p is a low-power packet then
    table[n,1] = true
else
    // p is a data packet
    // check whether it is on the ring of
       node n, i.e. has received high-power
       packet and has not received
       low-power packet from n
    if table[n,0] == true and table[n,1] == false then
        if the routing algorithm specifies to forward this
         packet then
            if ctrlSent == false then
                // has not yet sent high-power
                   and low-power packets
                schedule to send high-power packet
                schedule to send low-power packet
                ctrlSent = true
            end
            schedule to send the data packet p
        else
            discard the packet p
        end
    end
end
```

IV. Evaluation of the ring scheme

We evaluate our scheme in a nanonetwork, defined as a network of nanonodes [4]. Being very tiny, these nodes can be used create ultra-dense networks. Nanonodes are envisioned to communicate in the terahertz band (0.1–10 THz), using a specific ultra wide-band (UWB) modulation, Time Spread On-Off Keying (TS-OOK) [4], where a bit 1 is sent as a femtosecond-long ($= T_p$) pulse with energy, while a bit 0 is a silence (no energy). The ratio between the inter-bit duration T_s and the duration of one bit T_p is known as time spreading ratio (β). We use the standard values for them: $T_p = 100\,\text{fs}$ and $\beta = 1000$ [4]. Their is no explicit mechanism to avoid or detect collisions as it is not required here; nodes access the channel whenever they need to and rely on the inherently low probability of collisions on the very wide channel.

A. Available simulation software

Our study targets ultra-dense networks (thousands of nodes, with hundreds of neighbours per node) and thus requires a highly scalable network simulator. *ns3* network simulator has two different nanonetwork plug-ins: Nano-Sim [18] and TeraSim [19], but, due to a relatively heavy ns3 footprint, they are not scalable (up to around one thousand nodes).

BitSimulator [20] is a wireless nanonetwork simulator that supports both routing and transport levels. It can simulate tens of thousands of nodes in a network on a classical laptop. It also comes with a very useful visualisation tool, VisualTracer, that displays the simulation results, in particular node states (sending, receiving, collision, etc.) Therefore, we use BitSimulator to evaluate our ring scheme.

B. Scenarios

The simulation parameters are shown in Table I. The network is a square area. It has (a) 10 000 nodes or (b) 20 000

TABLE I: Simulation parameters.

Size of simulated area	6 mm * 6 mm
Number of nodes	10 000 or 20 000
Communication range	900 μm
Range1	900 μm
Range2	800 μm
Data packet size	1003 bit
Control packet sizes	101, 102 bit

nodes, distributed in an area of 36 mm^2; such a density can be found in software-defined metamaterials and in in-body communication. The nodes are placed randomly, using a uniform distribution, and are static. Nodes have omnidirectional antennas and equal communication ranges CR (except for the particular case of control packets).

The ring, i.e. the difference between areas of two communication ranges, is highly influenced by the radio propagation model. In the unit disc model, the ring is perfect, as the area between two concentric circles. In real world however, the communication ranges are not perfect circles. Thus, we implemented a *shadowing* propagation model, where packets are always received at distance $[0, d]$ from transmitter, received randomly in $]d, CR]$ with decreasing probability from 1 to 0, and lost at distance $>CR$ (communication range). The shadowing causes nodes in the ring to not receive control1 packets sometimes, and this makes the ring area as two blurred concentric circles. It also causes reception inversion: nodes inside the small circle, near its circumference, receive the control2 packet, but not the control1 packet.

The ring is specified by two radiuses, range1 and range2. We chose range1 to be equal to the communication range (so that all the nodes can receive it), and range2 smaller (so that it is received only by nodes outside the ring). The ring width is the difference between the two ranges, $w = 900 - 800 = 100\,\mu$m, i.e. $100/900 \approx 11\%$ of the communication range.

For the example scenario used, if we had used the unit disc model, for 10 000 nodes, the average number of neighbours per node would be 616, with a minimum of 167 (for corner nodes) and a maximum of 775. The average number of *ring* neighbours per node would be 122. The 20 000 nodes scenario has 1234 neighbours per node and 259 *ring* neighbours, in average. Using the shadowing propagation model, the average number of neighbours per node is 408 for 10 000 nodes and 819 for 20 000 nodes. This clearly represents an ultra-dense network.

The two control packets have random payload data, and are smaller than data packets. Distinctive values are chosen for data packet size (1003 bits) and control packet sizes (101 and 102 bits) simply to be easily spotted in the simulation log files.

A given source node (at the middle-top of the networks) generates a CBR flow of 10 packets (or 100 packets in one case) to all the nodes (in flooding) or to a node at the bottom-right of the network (in unicast). The number of hops is $x/CR = 6/.9 \approx 6.7$ hops on each dimension of the network, which is sufficient for testing routing protocols. The interval of time between two consecutive packets is large enough to avoid collisions between them. Given that each node sends the two control packets only once, right before the very first data packet it forwards, the effect of injecting the additional control packets should be reduced for a 10-packet flow.

As already mentioned, the aim of adding the ring to routing protocols is to restrict the forwarders to the ring area at each hop. We show the ring effect on two protocols: a flooding one, and a unicast one (Stateless Linear-path Routing SLR) [21].

We present in this article the values of more than 80 simulations: 2 network sizes (10 000 and 20 000 nodes) * 2 routing protocols * 2 scenarios per protocol (with and without ring) * $\geq$ 10 runs. Except if otherwise stated, each point of simulations shown represents the average of the 10 runs, differing only in the node positions (via a different seed of the random number generator dealing with node placement exclusively). However, for simplicity, we present detailed information for the 10 000 node scenario, and at the end a short comparison with the 20 000 node scenario.

We present the results using several metrics. We reiterate that our method aims to identify better forwarders, which should reduce the number of total forwarders, without impacting the delivery rate. Hence, the first main metric is about *forwarders*, shown as the cumulative number of senders for each packet, averaged for each of the 10 packets and over the 10 runs. The second main metric is about *delivery rate* (number of receivers): in flooding, the cumulative number of receivers for each packet, averaged for each of the 10 packets and over the 10 runs; in SLR, just whether the packet reached the destination (1) or not (0), averaged for each of the 10 packets and over the 10 runs. The third metric measures the traffic incurred by the two additional packets (control1 and control2). Finally, we show the main finding for the *second scenario* (with 20 000 nodes).

To desynchronise node forwarding in an ultra-dense network and reduce collisions, nodes chose a random delay (backoff) before forwarding any data or control packet, from a fixed window of 10 000 fs in SLR, and from a dynamic window in backoff flooding that is a function of the number of neighbours and sufficiently large for the node to count packet copies, as explained in [22].

To avoid forwarding loops, nodes forward packets they receive for the first time. For this to work, nodes can memorize the source ID and the packet sequence number of the received packets. This list is relatively small because a packet does not stay too much time in the network, hence nodes can safely remove old packets.

To have an insight on the reduction in the number of forwarders, and to increase the trust on the results, we show table results as average values of 10 runs with 10 packets each. The figures showing visually the network are captured from VisualTracer and show all the cumulative events per packet that occurred in the network with two key colors for nodes: blue for senders and green for receivers (we choose to show the first packet for the first run only as an example). Technical details and information about full reproducibility of our results

TABLE II: Results with 10 000 nodes.

	Without ring	With ring
Backoff flooding:		
Average number of forwarders per packet	100	72
Average number of receivers per packet	9 999.99	9 999.7
SLR:		
Average number of forwarders per packet	406	84
Destination reached	100%	99%

are provided on a separate web site[1].

C. Effect of the ring on backoff flooding

We want to check the ring effectiveness on an already efficient propagation scheme. Backoff flooding [22] is a highly efficient variant of flooding, which automatically chooses a very small number of forwarders (no matter the network density), and avoids the use of probabilities. In backoff flooding, upon reception of a packet, a node chooses a random backoff inside the waiting (contention) window. The window for each node is proportional to its number of neighbours. For that to work, nodes need to know the number of their neighbours, using DEDeN [23] for example. At the end of the backoff time, if the node has received less than n copies of the message, it forwards the packet, otherwise it discards it without forwarding. In this scenario, the number of copies chosen n is 1.

Adding the ring to the backoff flooding involves two modifications. The first is selecting only the neighbours in the ring to forward packets. The second is on the waiting window size. We recall that the time window is proportional to the number of forwarders; in the original backoff flooding, this is the number of neighbours, but when using ring this is the number of *ring* neighbours. Thus, the waiting time is modified to be proportional to the number of ring neighbours.

Table II shows that the ring reduces the number of forwarders by $1 - 72/100 = 28\%$, while the delivery ratio is similar (9999.7 vs 9999.99). We expect that the number of forwarders will be drastically reduced.

Fig. 1 shows the cumulative distribution of the number of forwarders in time, as a percentage of the total number of nodes in the network. Again, the progression for both without and with ring is regular, e.g. there is no sharp increase and the without ring curve increases much more compared to with ring curve.

The results show that the ring further increases the efficiency of an already highly efficient backoff flooding.

D. Effect of the ring on SLR routing

SLR (Stateless Linear-path Routing) [21] is an addressing and *unicast* (or merely *zone-cast*) routing scheme (i.e. message is sent to one destination only) designed for nanonetworks. In the *setup* phase, a few anchor nodes broadcast a packet in the network with a *hop* field set to 0, forwarded by all the nodes after increasing the hop count, whose effect is to divide the network space in *zones* and allow nodes to set their coordinates

[1]http://eugen.dedu.free.fr/bitsimulator/wcnc22

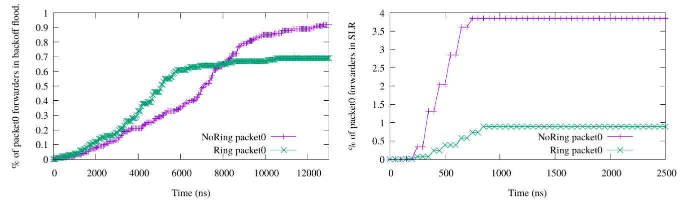

Fig. 1: Cumulative number of forwarding nodes over time in backoff flooding (left) and SLR (right) without and with ring, 10 000 nodes.

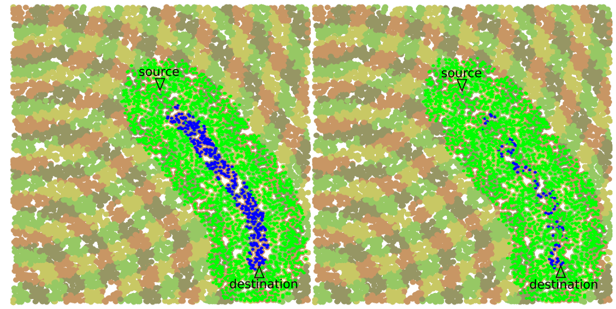

Fig. 2: SLR without (left) and with the ring (right).

as the hop count to each of the anchor. From that time on, upon reception of a packet to *route*, nodes forward it if and only if they are on the line between the source and the destination.

Adding the ring to SLR simply restricts the new forwarders to be in the ring at each hop, as shown in Fig. 2 right, i.e. the forwarders are found towards the end of each SLR zone, and fewer than without ring.

The results in Table II show that the ring reduces the number of forwarders by $1 - 84/406 \approx 79\%$, while the destination at bottom right still receives the message. Fig. 1 shows the cumulative distribution of the number of forwarders in time, as a percentage of the total number of nodes in the network. Again, the progression for both without and with ring is regular, e.g. there is no sharp increase and the without ring curve increases much more compared to with ring curve. To conclude, SLR with ring performs better than without ring.

E. Influence of the ring on the traffic

Compared to without ring, the additional two control packets generate additional traffic. However, we emphasize that the control packets for the ring creation are generated at most once by each node in static conditions.

In the scenario chosen, the data packet has 1003 bits, and the two control packets have 101 and 102 bits (cf. Table I). Given that the ring leads to a significant decrease of the number of forwarders and hence the number of sent packets, we expect that the ring will reduce the data traffic. Table III (last column) confirms that the number of packets sent with ring is highly reduced compared to without ring. To conclude, the benefit in reduction of packets generated outweighs by far the additional size of control packets, hence the use of the ring reduces the network traffic. Note that the two control packets may add an additional small transmission delay only for the first packet generated by nodes.

TABLE III: Total data traffic generated for 10 000 nodes.

	Without ring	With ring	Reduction
Backoff flooding	1 012 kbits	861 kbits	19%
SLR	4 079 kbits	905 kbits	77%

TABLE IV: Evaluation results for 20 000 nodes.

	Without ring	With ring
Backoff flooding:		
Average number of forwarders per packet	103	73
Average number of receivers per packet	20 000	19 999.9
SLR:		
Average number of forwarders per packet	909	155
Destination reached	100%	100%

F. Evaluation of ring in denser networks

All the previous results have been obtained in a scenario with 10 000 nodes. For a better confidence in the results, we evaluate the ring in another, denser, network. We keep all the parameters identical, except the number of nodes, increased to 20 000 nodes. The full results are shown in Table IV.

The main finding of these results when comparing to previous scenario (10 000 nodes) is that the reduction in the number of forwarders when using the ring as compared to without the ring is generally even higher: for backoff flooding, the reduction is $1 - 72/100 = 28\%$ with 10 000 nodes and $1 - 73/103 = 29\%$ with 20 000 nodes. For SLR with 10 000 nodes the reduction is 79%, and for 20 000 nodes it is 82%. The average delivery ratio is giving 1 or close to 1 values. Thus, the benefit of the ring increases in a denser network.

V. Conclusion and future work

This paper presented a new method to select forwarders in a multi-hop transmission. Using two control packets transmitted once (only before the very first data packet of a forwarder) at different power levels, a ring of nodes near the communication range of a transmitted node are selected as forwarders.

We validated the ring scheme in a simulator with two protocols: a optimized flooding and a destination-oriented protocol. We compared the results using several metrics: number of forwarders, delivery ratio and data traffic, and in two different network node densities of 10 000 and 20 000 nodes. The ring achieves much better performance in both protocols, and for the main metrics (number of forwarders, and delivery ratio), with slightly lower results for the data traffic. The performance increases with the network density.

Future work includes a theoretical study, dynamically selecting the ring width according to node density, mobile nodes, and heterogeneous networks.

Acknowledgments

This work has been funded by Pays de Montbéliard Agglomération (France).

References

[1] A. James, A. Madhukumar, and F. Adachi, "Efficient multihop transmission scheme for error-free relay forwarding in cooperative networks," *Wireless Communications and Mobile Computing*, vol. 15, no. 4, pp. 755–769, 2015.

[2] L. Kleinrock and J. Silvester, "Optimum transmission radii for packet radio networks or why six is a magic number," in *National Telecommunications Conference*, Piscataway, NJ, USA, Dec. 1978, pp. 4.3.1–4.3.5.

[3] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Networks*, vol. 8, no. 2–3, pp. 153–167, Mar. 2002.

[4] J. M. Jornet and I. F. Akyildiz, "Femtosecond-long pulse-based modulation for terahertz band communication in nanonetworks," *IEEE Trans/ on Communications*, vol. 62, no. 5, pp. 1742–1753, May 2014.

[5] F. Lemic, S. Abadal, W. Tavernier, P. Stroobant, D. Colle, E. Alarcón, J. Marquez-Barja, and J. Famaey, "Survey on Terahertz Nanocommunication and Networking: A Top-Down Perspective," *IEEE Journal on Selected Areas in Communications*, pp. 1–37, Apr. 2021.

[6] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "A new wireless communication paradigm through software-controlled metasurfaces," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 162–169, Jun. 2018.

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

[8] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized link state routing protocol (OLSR)," INRIA, Le Chesnay, France, Tech. Rep. 5145, Mar. 2004.

[9] L. Maccari, M. Maischberger, and R. L. Cigno, "Where have all the MPRs gone? on the optimal selection of Multi-Point Relays," *Ad Hoc Networks*, vol. 77, pp. 69–83, 2018.

[10] C. Liang, S. Lim, M. Min, and W. Wang, "Geometric broadcast without GPS support in dense wireless sensor networks," in *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*. IEEE, 2014, pp. 405–410.

[11] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, "Contention-based forwarding for mobile ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, 2003.

[12] M. Witt and V. Turau, "BGR: Blind geographic routing for sensor networks," in *Third International Workshop on Intelligent Solutions in Embedded Systems*. IEEE, 2005, pp. 51–61.

[13] Y. Cai, K. A. Hua, and A. Phillips, "Leveraging 1-hop neighborhood knowledge for efficient flooding in wireless ad hoc networks," in *IEEE International Conference on Performance, Computing and Communications (IPCCC)*, Phoenix, AZ, USA, Apr. 2005, pp. 347–354.

[14] B. Karp and H.-T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *6th Annual Int. Conf. on Mobile computing and networking (MobiCom)*, Boston, MA, USA, Aug. 2000, pp. 243–254.

[15] M.-T. Sun, W.-C. Feng, T.-H. Lai, K. Yamada, H. Okada, and K. Fujimura, "GPS-based message broadcast for adaptive inter-vehicle communications," in *IEEE Conference on Vehicular Technology (VTC)*, vol. 6, Boston, MA, USA, Sep. 2000, pp. 2685–2692.

[16] S. Panichpapiboon and L. Cheng, "Irresponsible forwarding under real intervehicle spacing distributions," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2264–2272, Jun. 2013.

[17] D. Rossi, R. Fracchia, and M. Meo, "VANETs: Why use beaconing at all?" in *IEEE International Conference on Communications (ICC)*, Beijing, China, May 2008, pp. 2745–2751.

[18] G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Nano-Sim: simulating electromagnetic-based nanonetworks in the network simulator 3," in *6th International ICST Conference on Simulation Tools and Techniques (SimuTools)*. Cannes, France: ACM, Mar. 2013, pp. 203–210.

[19] Z. Hossain, Q. Xia, and J. M. Jornet, "TeraSim: An *ns-3* extension to simulate Terahertz-band communication networks," *Nano Communication Networks*, vol. 17, pp. 36–44, Sep. 2018.

[20] D. Dhoutaut, T. Arrabal, and E. Dedu, "BitSimulator, an electromagnetic nanonetworks simulator," in *5th ACM Int. Conf. on Nanoscale Computing and Comm. (NanoCom)*, Reykjavik, Iceland, Sep. 2018, pp. 1–6.

[21] 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.

[22] T. Arrabal, D. Dhoutaut, and E. Dedu, "Efficient multi-hop broadcasting in dense nanonetworks," in *17th IEEE International Symposium on Network Computing and Applications (NCA)*. Cambridge, MA, USA: IEEE, Nov. 2018, pp. 385–393.

[23] ——, "Efficient density estimation algorithm for ultra dense wireless networks," in *International Conference on Computer Communications and Networks (ICCCN)*, Hangzhou, China, Jul. 2018, pp. 1–9.