Lightweight, Self-tuning Data Dissemination for Dense Nanonetworks

A. Tsiliariadou\textsuperscript{a}, C. Liakos\textsuperscript{a,\textdagger}, S. Ioannidis\textsuperscript{a}, A. Pitsillides\textsuperscript{b}

\textsuperscript{a}Foundation of Research and Technology-Hellas, Heraklion, Greece
\textsuperscript{b}Department of Informatics, University of Cyprus, Nicosia, Cyprus

Abstract

A nanonetwork comprises a high number of autonomous nodes with wireless connectivity, assembled at micro-to-nanoscale. In general, manufacturing and cost considerations imply that nanonetworking approaches should have minimal complexity, ideally without sacrifices in network coverage. The present paper studies a networking approach fit for static, dense topologies comprising numerous, identical, computationally-constrained nodes. These attributes are especially important in the context of recently proposed applications of nanonetworks. The presented networking approach assumes that each node is equipped with 10 bits of reclaimable storage to accommodate four integer counters, and a trivial set of integer operations on them. These modest resources are used for logging packet reception statistics. Nanonodes with good reception serve as retransmitters within the network. This classification process is based on the Misra-Gries algorithm, used for detecting frequent items in sequential streams. Evaluation via extensive simulations in various 2D and 3D topologies yields high network coverage, achieved with less resources than related approaches.

Keywords: Wireless Networks, Nanoscale, Low Complexity, Dense Topologies.

1. Introduction

Recent advances in nanotechnology enable the creation of nano-sized power units, antennas, communication modules and CPUs, promoting the advent of nanonetworking [1]. Swarms of autonomous, wireless nodes that can sense and act on their environment will introduce radical changes in everyday life and the industry [2]. Presently, however, manufacturing autonomous nodes at nanoscale implies extremely limited hardware capabilities [1]. Furthermore, interesting applications of nanonetworks may require thousands to millions of nodes [2, 3, 4], posing cost issues as well. Thus, nanonode architectures must be as simplistic

\textdaggerCorresponding author
Email addresses: atsioli@ics.forth.gr (A. Tsiliariadou), cliaskos@ics.forth.gr (C. Liakos), sotiris@ics.forth.gr (S. Ioannidis), cpsiti@cs.ucy.ac.cy (A. Pitsillides)
as possible, while fulfilling efficiently their application-specific goal. Given that the networking module is but a part of the nanonode architecture, it should require even fewer resources in terms of complexity, without compromising the connectivity of the nanonetwork.

The present study proposes a nanonetworking model fit for static, dense networks with numerous nodes. An important assumption is that the nodes are unique in all hardware and geometry aspects. Such networks can be used for the nano-scale monitoring of mission-critical materials, e.g. embedded within nuclear reactors [5]. Another important application is in the context of the recently proposed Software-Defined Materials (SDMs) [4]. SDMs allow for programmatic control over the electromagnetic (EM) behavior of an object. Their functionality stems from metamaterials, a class of artificial materials with a well-defined, periodic structure. SDMs equip a standard metamaterial with a network of nanonodes, which can receive external commands and alter the internal structure of the metamaterial accordingly. Thus, the EM properties of an object as a whole can be controlled programatically as well.

The nano-scale allows for very high concentrations of nodes per volume, compared to macro-scale networks. The handling of dense nanonetworks yields three main challenges. Firstly, the scalability in terms of manufacturing and node communication is a major issue. For example, certain SDMs may require $10^{-100}$ nodes per $mm^3$ [4]. Therefore, the manufacturing cost per node should be minimal, implying a very simple hardware architecture. On the other hand, overly simplified architectures may not even support simple communication protocols, yielding connectivity issues as the number of nodes increases [6]. Secondly, the networking paradigm should offer high coverage per packet, with as few retransmissions as possible. Minimal packet retransmissions favor energy-efficiency and packet delivery time, since redundant transmissions and communication errors are minimized. Thirdly, the communication channel model within dense nanonetworks has a unique attribute: the nodes themselves act as non-trivial obstacles to the propagating EM wave [7].

The present paper contributes a networking scheme that addresses these challenges as follows. Firstly, a flood-based communication paradigm is adopted. Thus, the node architecture is simplified, since no medium access or routing protocol is required. This results into manufacturing cost benefits, and highly scalable node communication offered by the flood paradigm [8]. Secondly, packet retransmissions are reduced significantly with regard to related approaches. The methodology is based on the real-time classification of the nodes into few retransmitters (“infrastructure” nodes) and many passive auditors (“user” nodes), depending on their packet reception statistics. Internally, each node maps incoming packets to reception outcomes (e.g., successful or failed). The formed sequence is processed by a novel variant of the Misra-Gries algorithm [9], which detects the most frequent outcome types. The output of the algorithm deduces the classification result. This process has an extremely small computational footprint, while requiring trivial, integer processing capabilities only. Thirdly, the proposed scheme is evaluated in a state-of-the-art simulator that employs 3D ray-tracing to approximate the EM propagation between all node pairs within
the nanonetwork [10]. Statistical channel models are shown to diverge significantly from the ray-tracing results in the studied dense topologies.

The remainder of this paper is organized as follows. Related studies are given in Section 2. The system model and application context are given in Section 3. Section 4 details the proposed networking scheme. Evaluation via simulations takes place in Section 5. The conclusion is given in Section 6.

2. Related work

Nanoneering is presently studied from two different angles. The first approach proposes biological or bio-inspired communication modules. For example, the nodes may encode a single piece of information on several biological molecules (e.g., RNA) and diffuse them to their environment, or may exchange data upon collision only, mimicking the operation of viruses [2, 11]. The second approach, which is assumed in the present study, considers wireless EM communication [1]. Related studies have so far focused on defining operational physical (PHY) layer specifications and Medium Access Control (MAC) protocols. We will first provide a brief overview of the PHY layer of nanocommunications, highlighting its challenges and the assumptions of the present work. Then, we will focus on MAC approaches which are related to the present work.

PHY layer. Related studies focus on specifying the channel model, nanoantenna geometry, and the power supply module. Concerning the channel model and nanoantennas, studies support that the most promising operating spectrum is the Terahertz Band (0.1 – 10.0 THz) [12]. The miniaturization of the antennas at nanoscale, while keeping the operating frequency tractably high (THz) can be achieved with the use of graphene. More particularly, the propagation speed of EM waves in carbon plasmonic nano-antennas can be orders of magnitude lower than in classical materials, yielding antennas 100 – 1000 smaller than conventional ones for the same wavelength [2]. Recent studies have shown that the communication range of a single node may be increased with the use of the 0.1 – 0.54 THz window [13]. The authors showed that, when using this window, the free-space propagation loss becomes the dominating factor in channel characteristics, minimizing molecular absorption and achieving the largest transmission distance. Particularly, the 100 GHz is very promising, assuming standard atmospheric conditions, since it corresponds to a local minimum in terms of molecular absorption, while still offering high data rates [14]. However, it has been recently shown that even small objects, such as aerosol particles, affect the channel model substantially, introducing additional EM scattering [7]. Nonetheless, the nanonodes themselves have not been considered in any channel model so far, despite being themselves substantial EM scatterers, especially in dense topologies. Regarding modulation and encoding, Jornet at al. [15] proposed the use of Rate Division Time Spread On-Off Keying (RDTS-OOK). Nanonodes transmit a logical "1" as a femtosec-long pulse and a logical "0" as silence. Any two nodes choose a unique inter-pulse time interval via a handshake protocol, in order to minimize collisions. In absence of handshaking collisions are possible, considering the wide spectrum of a femtosec pulse. Finally, the
nanonode power supply is perhaps the most challenging module. Jornet et al. propose an energy scavenging unit based on piezoelectric nanogenerators, which can store 800 µJ of energy in 50 sec, occupying an area of 1000 µm² [1]. Given that a 25 byte-long packet consumes 200 µJ (RD TS-OOK) [1], a maximum of 1 packet per 12.5 sec can be sustained via energy-scavenging, which can pose a problem even for simple handshake-based communication protocols. Similar results are reported by the solution of Mohrehkesh et al., with a harvesting module that can store 20 µJ with an average rate of 0.5 µJ/sec, and a drain of 2 – 10 µJ per transmitted packet [16]. Thus, 1 packet exchange per 10 sec can be sustained as well. Wang et al also report a solution that can store up to 9 nJ with an average rate of 63 µJ/sec [17].

Apart from energy-harvesting, the present work also considers a power supply based on wireless power transfer (WPT) [18]. WPT solutions can be categorized as either radiative or non-radiative [19]. In the radiative case, source-antennas emit electromagnetic power in the form of propagating waves to power-consuming devices located at a far distance. The source-antenna needs to be sufficiently powered to account for the path loss. The non-radiative WPT mainly relies on the near-field coupling between two inductive coils and has been popularized by RFID chips [20] among other applications [21]. Particularly, we highlight the RFID “dust” by Hitachi Ltd., whose nodes are manufactured at a total size of 50×50×5 µm, i.e., including all components [22]. WPT offers a much more stable-albeit still limited-power supply, allowing for packet transmission at sub-sec time, enabling applications such as highly responsive Software-Defined Materials [4]. The disadvantage is that the network is active only while the external power supply is on.

MAC Layer. Related approaches have so far studied sparse dynamic topologies and assorted protocols, mainly targeting Body Area Network (BAN) applications [23]. Furthermore, most studies assume hierarchical networks, where a set of µm-sized, relatively powerful nano-routers control the µm-sized, weaker, cheaper nanonodes [24]. It is noted that the hierarchical approach tends to be more intrusive, in the sense that it requires the presence of sizable nanorouters. Thus, it may not be fit for some applications apart from BANs. For example, the structure and properties of a monitored material may be significantly altered with such a nanonetwork residing within. Scalability and deployment are important issues as well. Wang et al. proposed a MAC protocol for hierarchical nanonetworks, using a scavenging-based power supply [17]. The protocol is handshake-based and targets fair channel access and throughput per node. A critical packet transmission is derived below which the network can operate perpetually. The same goal is pursued by Jornet et al. [1] and Pierobon et al. [24], using detailed, analysis-supported power scavenging models. Afsharinejad et al. propose a hierarchical, handshake-based approach for plant monitoring, which employs FDMA and performs frequency channel hoping to reduce the probability of collisions [23]. A similar approach is employed by D’Oro et al. [26] and Srikanth et al. [27], which use TDMA instead. Mohrehkesh et al. propose the Receiver-Initiated Harvesting-aware MAC protocol (RIH-MAC). RIH-MAC reverses the sender-oriented logic of CSMA/CA-CD and proposes the
Ready-to-Receive (RTR)/DATA signals instead. Once a receiver has the proper
energy level and channel conditions to receive a data, it broadcasts an RTR
signal to all interested senders. The senders reply with a DATA packet that
conveys the useful information. RIH-MAC is fit for sparse topologies in order
to avoid saturating the network with RTR signals.

Very few studies consider ad hoc networks of identical nodes. Jornet et
al. proposed PHLAME, a distributed MAC protocol running on top of RD-
TS-OOK, which allows a transmitter and a receiver pair to choose the optimal
communication parameters on demand, through a lightweight handshaking pro-
cess [15]. RIH-MAC also has an ad hoc variation that operates as follows [6].
First, the nano-nodes exchange probing packets to perform topology discovery.
Then, the link sharing is mapped to the link-coloring graph problem, which is
NP-Hard. The algorithm of Grable and Porconisi algorithm is run locally on
each node in order to solve the problem by approximation [28].

Conceptual similarities also exist among the studied nanonetworks and ad
hoc networks-on-chips (NoCs) [29]. Particularly, self-organized NoCs need to
discover their topology and perform defect mapping in a completely decentral-
ized manner [30]. Looping paths among chips is another major concern, which
can be mitigated in various proposed ways, such as virtual channels [31] and
spanning trees [32]. Similar research topics have also been studied in the con-
text of classic, macro-scale Wireless Sensor Networks (WSNs) [33]. Nonetheless,
NoCs and WSNs assume much more powerful nodes than nanonetworks, even
able to support a full protocol stack [29, 33]. Therefore, WSN and NoC-oriented
solutions are generally not applicable to nanonetworks. The reader is directed to
the work of Mohrehkesh et al. for an overview of the unique problems faced even
at the level of rudimentary communication establishment among nodes [6]. We
note that even nanonode addressing is an open problem, among other nanone-
working issues [34].

**Differentiation.** In terms of PHY layer, the present study considers the use
of wireless power transfer for nanonodes. This enables new classes of applica-
tions (e.g., [4]) which were previously not feasible via energy scavenging (which
supports transmitting approximately 1 packet per 10 sec per node). The wire-
less power transfer does not make for unlimited or uninterrupted energy supply.
However, it constitutes a much more stable and dependable source compared
to scavenging-based solutions. Energy scavenging is also discussed, for compat-
ibility with the existing body of related work. Furthermore, the simulations
performed in the context of the present work consider the EM scattering ef-
effects of the nanonodes themselves, apart from molecular absorption and fading
effects. To the best of the authors knowledge, this approach is unique to the
present work. A ray-tracing engine is employed for the task, which has been
validated with accurate Finite-Differences Time-Domain (FDTD) methods [10].
In terms of topology, the present work assumes dense topologies of numerous
nodes, in contrast to the sparse, small topologies that have been considered so far.
Furthermore, implementation cost concerns are introduced to the design of a
nanonetwork.

The cost and scalability concerns are addressed by simplifying the commu-
communication paradigm of the nanonodes and considering identical nodes only (i.e., non-hierarchical, ad hoc nanonetworks). Specifically, a flood-based approach is adopted, abolishing the need for handshaking procedures. The packet retransmissions are minimized by limiting the number of nodes that retransmit packets via a real-time node classification scheme. The classification uses node-local measurements of packet reception statistics. The prior work of Liaskos et al. [35] focused on proving analytically that such an approach can reduce significantly the redundant packet retransmissions, without segmenting the network. However, implementation on low-cost, computationally weak, or energy-constrained nodes was not discussed. The proof-of-concept simulations of the same work assumed floating point computation capabilities per nanonode, real-time signal processing capabilities at picosec scale, and no concern for classification runtimes. In contrast, the present work focuses on realistic implementation requirements. It requires trivial integer processing capabilities, no signal processing requirements, 10 bits of memory in total, while the classification is concluded in minimal times ($n$ sec).

3. System model and application context

Physical-layer assumptions. The present study assumes the system model of Fig. 1. The nanonodes are identical, modeled as cubes with a side of 10 $\mu$m [36]. Their architecture, given in Fig. 1a, comprises a trivial $\mu$CPU, a wireless communications component, a power supply and a sensor/actuation module. The $CPU$ is considered to perform integer calculations only and sustain a few bytes of memory. The wireless communications module operates at 100 GHz, for the reasons detailed in Section 2. This frequency corresponds to a wavelength of 3 $mm$. Therefore, a graphene antenna is required, given that its size can be $10^3$ times smaller than the operating wavelength [37]. Two approaches are assumed
for the nanonode power supply. The most prominent one is based on wireless power transfer [19], while energy-scavenging and nanobatteries are also discussed as a secondary solution (cf. Section 4.2).

Following the inductive coupling approach for WPT, we assume that each nanonode is equipped with one or more µm-sized coils [22], as shown in Fig. 1a. These µcoils are coupled to an external, larger coil, cf. Fig. 1b. The coupling coefficient depends on the surface of the coils, their number of loops, their distances and relative angles [18]. Thus, equipping each side of the cubic nanonode with a coil increases the received power. In addition, notice that a 10µm coil acts as an antenna at a wavelength of \(~20\) m, limiting the interference to the wireless communications module. We note that the metamaterial technology can also be used for implementing the µcoils. For example, authors in [38] proposed a µm-sized Edge-Coupled, Split Ring Resonator, which was used for coupling nanonode pairs with 90% efficiency at 7.4 – 7.6 THz. Further metamaterial-inspired coil designs for WPT applications can be found in [39, 40].

Communication model. In our system, the data communication between the nanonetwork and the external world is bidirectional and flood-based (Fig. 1b). An external entity may send a data packet (e.g., containing commands), which must be propagated to the nanonetwork as a whole (broadcast), with high coverage in terms of percentage of informed nodes. A node may also send data towards the external world (e.g., sensing data). Notice that the external entity has no fixed position and, therefore, each nanonetwork-originated packet must be delivered to all sides of the host-object (i.e., with high coverage as well).

Due to the high-coverage requirements, the data routing model follows a selective retransmission paradigm. According to it, the nodes are self-classified in real-time as retransmitters or passive auditors. The nodes classified as retransmitters then blindly forward all incoming packets (flood), while the auditors do not participate in the retransmission process. This classification process has been shown to exhibit some strong properties [35]. First, each node classifies
Figure 3: Periodic arrangements of nanonodes can be used to create materials with programmable electromagnetic properties.

itself based on personal receptions statistics, without any form of direct interaction with their neighborhood. Secondly, very few nodes get classified as auditors, especially in denser networks, promoting energy efficiency. Thirdly, the classification is online and adaptive, meaning that new nodes can assume the roles of retransmitters once the old ones fail (e.g., run out of energy). Finally, the nodes that get classified as retransmitters tend to form well-defined patterns. An example is given in Fig. 2, where the center-most node senses data and seeks to transmit it to an external entity. The nanonodes are self-classified as shown in the same Figure. The ray-like formation, which appears even in random topologies, can be effectively used for pinpointing the location of a sensing event, just by performing power measurements around the nanonetwork area [35]. In addition, packets are forced to follow straight paths (i.e., over the rays) promoting low delivery times.

It is clarified that none of the aforementioned benefits require a node addressing scheme, providing an alternative to this major research issue [34]. The adopted communication model assumes no node identifiers, and yet is able to pinpoint the location of sensing events, as discussed. We note, however, that while the adopted model is flood-based, point-to-point protocols can be built on top of it. For instance, the nodes classified as retransmitters can operate as a bus, conveying packets exchanged between a single pair of “auditor”-classified nodes. Nonetheless, such an approach is beyond the scope of the present work.

Application context. The proposed nanonetworking approach targets: i) the real-time monitoring of objects, and ii) software-defined materials [4]. In the real-time monitoring of objects, the nanonodes are equipped with a sensing module to derive a critical parameter of the material (e.g., forming structural deficits). The sensed data are transmitted to an external entity, which pinpoints the event location as discussed. In software-defined materials, each nanonode
Figure 4: State chart of a nanonode operating by the proposed approach.

controls a set of Micro Electro Mechanical switches (MEMS) [41]. The nanonode and its MEMS switches form a pattern unit, as shown in Fig. 3. An external entity sends a packet that defines the state of each switch. This packet is transmitted with high coverage within the hosting object. Each nanonode applies the directives of this packet, thus forming periodic patterns within the object. These periodic structures are essential a metamaterial [42]. The EM properties of a metamaterial stem from its periodic structure. For example, Fig. 3 shows the formation of rectangular split rings. The gap opening of the split rings define the refraction angle of an EM wave impinging on the metamaterial [42]. To understand the impact of this potential, we mention that a carefully designed stack of metamaterials, each with its own refraction angle, was able to render an object invisible to EM waves (cloaking) by gradually bending them around it [43, 44]. Different patterns yield different type of EM behavior (e.g., refract or absorb), while the size of the pattern defines the interacting EM frequency band. Nano-sized patterns allow for interaction with visible light. Finally, the nanonodes can also report functionality issues (e.g., malfunctioning switches) back to an external entity, in the same manner as in real-time object monitoring.

4. The proposed nanonetworking approach

We propose a nanonode classification scheme that operates at the level of packet statistics. The packet reception process is treated as a black box, ignoring all physical-layer attributes, such as power levels [35]. In any broadcast-based environment, a new incoming packet event ("PACKET_IN") may lead to three major outcomes. A "PARITY_ERROR" event represents the case when a packet has been received, but it has failed the integrity checks, such as having an erroneous parity bit. A "DUPLICATION_ERROR" occurs when a packet has been successfully received, having passed all integrity checks as well, but is discarded because its duplicate has been already received before. Such events are
Algorithm 1 The MG Classifier routine.

```plaintext
1: procedure MG_Classifier(event)
2:   if event == 'PARITY_ERROR' then
3:     n ← n + 1
4:     ctrpe ← ctrpe + 1
5:     ctrrs ← max{0, ctrrs − 1}
6:     ctrde ← max{0, ctrde − 1}
7:   else if event == 'DUPICATION_ERROR' then
8:     ctrde ← ctrde + 1
9:     ctrpe ← max{0, ctrpe − 1}
10:    ctrrs ← max{0, ctrrs − 1}
11:   else if event == 'RECEPTION_SUCCESS' then
12:     n ← n + 1
13:    ctrrs ← ctrrs + 4
14:    ctrpe ← max{0, ctrpe − 1}
15:    ctrde ← max{0, ctrde − 1}
16:   end if
17:   return
18: end procedure
```

common in broadcast-based environments [45]. A "RECEPTION_SUCCESS" event occurs otherwise. Each "PACKET_IN" event is thus mapped to an outcome, forming a sequence. The methodology for deducing a node’s ability to serve as infrastructure is to find the most frequent items in this sequence. To this end, we employ a simplification of the Misra-Gries (MG) algorithm, which finds all frequent items in a stream, using positive integer counters only [9].

The original MG algorithm extracts the exact set of frequent items in a stream of $m$ items, using only $k$ counters. A frequent item appears by definition more than $\frac{m}{k+1}$ times in the stream. The counters form an associative array indexed by the items, $ctr_{item}$, and are initialized to zero. For each incoming item, MG operates as follows. If $ctr_{item} > 0$, increase $ctr_{item}$ by one. Else, if the number of non-zero counters is less than $k$, set $ctr_{item} = 1$. Otherwise, decrease all non-zero counters by 1. The storage overhead of MG in $O(k)$.

In the case of the proposed MG-based Dynamic Infrastructure (MG-DI), the algorithm is adapted as follows. We assume three positive integer counters, $ctr_{rs}$, $ctr_{pe}$ and $ctr_{de}$ for the "RECEPTION_SUCCESS", "PARITY_ERROR" and "DUPLICATION_ERROR" outcomes respectively. We employ an additional counter, $n$, to monitor the total number of processed outcomes. The updated routine is denoted as Algorithm 1 and its application context in given at the nanonode state chart in Fig. 4.

Initially, all counters are nullified and a node acts as a blind retransmitter for all incoming packets. On each "PACKET_IN" event, the MG_classifier routine is called and the counters are updated. When the total number of processed outcomes $n$ surpasses 2, the classification occurs based on a simple comparison between the current $ctr$ values. If the node has a good record of successful receptions compared to failures and duplications ($ctr_{rs} \geq ctr_{pe} + \Delta$), then the node is designated as a possible infrastructure candidate. Otherwise, it is classified as a failed packet counter or a failed duplication counter.
$ctr_{de}$), it deduces that it should act as a retransmitter. The fuzziness of the $(n > 2)$ condition expresses asynchronicity potential. The described process is not intended to block the overall operation of the node. The condition check must simply occur when $n$ is at least equal to 3, i.e., the first non-unary odd number, in order to avoid ties and minimize the scheme’s runtime.

Even if $ctr_{rs} \geq ctr_{pe} + ctr_{de}$ yields false, the node should still act as a retransmitter for a trivial equalization interval, in order to make sure that all nodes have deduced their classification as well. Thus, the classification of a node as a user/retransmitter does not affect the classification of its neighbors in general. In other words, if the equalization delay was removed and a node entered the “user” mode, the interference caused to its neighbors would decrease, encouraging them to turn into redundant “retransmitters”. A simple and effective way to implement the equalization interval is to wait until, e.g., $n > 10$, which is also inline with the integer computational capabilities assumption. Of course, real timers can be used as well. Notice that the user mode corresponds to a probabilistic retransmission in Fig. 4. This is applicable only to the energy scavenging-based operation, discussed in Section 4.2. In all other cases, the retransmission probability can be zero for the user mode.

It is noted that duplication errors are treated as indications for not assuming a retransmitter role, despite the fact that the reception of the packet was without other errors. This choice limits the probability of turning clusters of nodes into retransmitters, when, e.g., one or two would suffice. The rest of the nodes should better serve as a reserve pool, and assume the roles of retransmitters when required. This adaptivity is achieved by returning to the "INIT" from any classification state after a timeout ($OnTimeout$, Fig. 4). Therefore, the duplication errors have a secondary role, compared to the clearly positive "RECEPTION_SUCCESS" and the clearly negative "PARITY_ERROR" events. For this reason, the $n$ counter, which serves as a timer to the classification process, is advanced by the latter two, major events only.

**Remark 1.** The MG-DI scheme can deduce the classification of a nanonode within 3 packet processing events at a minimum, produced by any combination of "RECEPTION_SUCCESS" and "PARITY_ERROR" events only. In this boundary case, event triplets may contain 0 to 3 "RECEPTION_SUCCESS" instances. Given the error prone nature of packet transmissions in a flood-based nanonetwork, MG-DI ensures that a triplet containing at least one "RECEPTION_SUCCESS" event should lead to a retransmitter classification, i.e.,

$$ctr_{rs} \geq ctr_{pe} + ctr_{de} \quad \text{with} \quad ctr_{de} = 0 \implies ctr_{rs} \geq ctr_{pe}$$

To this end, the $MG$ _classifier_ routine introduces a non-unary increment step, $I = 4$, for the $ctr_{rs}$ counter at step 13. It is not difficult to show that the triplet leading to the minimum value of $ctr_{rs}$ starts with a "RECEPTION_SUCCESS" event, followed by two "PARITY_ERROR"s. Therefore, the final values of $ctr_{rs}$ and $ctr_{pe}$ are $I - 2$ and 2 respectively. Thus, from inequality (1) we deduce the optimal value of $I$ as:
4.1 Advantages over nature-inspired classifiers

\[ \text{ctr}_{rs} \geq \text{ctr}_{de} \Leftrightarrow I - 2 \geq 2 \Leftrightarrow I \geq 4 \]  \hspace{1cm} (2)

The strength of MG-DI is its extremely small footprint in terms of required computations and storage. Essentially, the proposed scheme operates with four, positive integer counters. Assuming that \( \max\{n\} = 3 \), the \( n \) counter needs just \( \lceil \log_2(3) \rceil = 2 \) bits. Likewise, the \( \text{ctr}_{rs} \) counter can be accommodated in \( \lceil \log_2(3 \cdot 4) \rceil = 4 \) bits, and the \( \text{ctr}_{pe} \) and \( \text{ctr}_{de} \) in \( \lceil \log_2(3 \cdot 1) \rceil = 2 \) bits each. Thus, the total storage overhead is 10 bits at a minimum. Given that MG-DI can deduce the classification result within 3 packet receptions, its minimum complexity is static and equal to \( O(3) \). We note that MG-DI requires increase/decrease and comparison operators on integers only. Finally, all storage can be reclaimed and used for other purposes once the classification is complete.

4.1. Advantages over nature-inspired classifiers

In the work of Liaskos et al. [35], the classification of nodes into users and infrastructure was performed by a nature-inspired classifier, the Dendritic algorithm [46]. The Dendritic algorithm mimics the way the human immune system classifies unknown objects (e.g., viruses) as threats. A Dendritic cell latches on an object, monitors associated biological "danger" and "safety" indications, and slightly increases or decreases its internal "alarm" level. If the observation of several objects concludes with the Dendritic cell being in "alarm" state, subsequent objects are treated as threats. By analogy, the Dendritic-Dynamic Infrastructure (D-DI) of [35] considers a Dendritic "process" running on each nanonode. The observed objects are the received packets, while the Useful-Signal and Interference-Plus-Noise power levels stand for the associated "safety" and "danger" signals. After a timeout, each node classifies itself as "infrastructure" or "user", based on its internal alarm state.

However, D-DI presents parametrization and implementation challenges. At first, the Dendritic algorithm requires mapping of "danger" and "safety" indications to Useful-Signal-Power (S) and Interference-Plus-Noise-Power (IN) waveforms. In other words, a nanonode must separate S from IN in real-time and for each incoming packet. Then, a reduction must be applied on each waveform, to produce single floating point values. The type of the reduction (e.g., min, max, mean) is an input parameter. In sparse topologies, keeping the maximal signal values is sufficient, while denser topologies may require the logging of average values. However, no standard rule exists for intermediate node densities. The described process may require additional hardware components and extra computational processing power per nanonode. Furthermore, the optimal number of packets to be processed prior to concluding the classification result, as well as the classification threshold, are also input parameters, imposed by the classic Dendritic algorithm [46]. The algorithm requires 4 float and 1 integer variables, plus floating point processing capabilities. Its complexity is \( O(x) \), \( x \) being the total packets needed to be processed prior to classification.
4.2 Energy scavenging-based operation

As discussed in Section 2, energy scavenging-based nanonodes have limited transmission potential. Approximately 1 packet may be transmitted per 10 sec at best, while a fully charged nanonode can transmit up to 4 packets before depleting its energy reserve. Given that the proposed networking scheme leads to node classification within 3 packet receptions, the 4-packet capacity does not pose an issue to the process. However, nodes classified as retransmitters will inevitably deplete their reserves soon. To better understand this effect we refer to Fig. 2. The retransmitters will handle the packet transmission load. Thus, nodes within the white ray-shaped areas will soon run out of energy.

In order to keep the proposed scheme operational in this case, the “user” nodes are requested to assist probabilistically in the packet retransmission process. In other words, the “user” nodes retransmit incoming packets with a very low probability. Thus, the network will exhibit two modes of operation. In the first mode, the white ray-shaped areas are powered on and the network operates as intended. The small number of retransmissions from user nodes does not alter this operation. The second mode occurs when the white ray-shaped areas have run out of energy. In this case, data dissemination is performed by the “user” nodes via the probabilistic packet forwarding. Once the “retransmitter” nodes are fully charged, the network reverts to the first mode once again.

The packet forwarding probability of “user” nodes is an input parameter in the case of energy harvesting-based operation. Notice that the optimal forwarding probability, which maximizes coverage and minimizes retransmissions, is approximately equal to the ratio of “retransmitters” [35]. In the studied, dense topologies the vast majority of nodes are classified as “users” [35]. Thus, the expected value of the optimal forwarding probability is very low (e.g., 5 – 15% indicatively). Finally, it is noted that the “user” forwarding probability can be set by an external entity via simple message broadcasting. Therefore, it can also be optimized via trial and error at the product design phase, given that the studied topologies are static.

Nonetheless, the limitations of energy scavenging in general may pose serious limitations to the targeted applications. For example, material monitoring may be possible only at a time period of 1 min. Likewise, the structure of the software-defined materials may be tuned in similarly large intervals, limiting the adaptivity potential to environmental changes.

5. Simulations

In this Section we evaluate the performance of the proposed MG-DI algorithm versus alternative solutions. The evaluation assumes a range of 2D and 3D topologies, node density cases and channel model configurations. The simulations were implemented on the AnyLogic platform [47]. The confidence of the presented results is 95%.

Where applicable, MG-DI is compared to the D-DI algorithm of [35], the OPTIMAL FLOOD [48] and a CSMA/CA-assisted flood approach [49]. According
to the optimal flood, each node retransmits a packet probabilistically. The lowest forwarding probability that achieves perfect coverage is considered optimal, and is derived from topological data [48]. Therefore, a separate optimization is needed for each topology/node density setup.

The CSMA/CA-assisted flood assumes that each node forwards a packet only when the wireless channel is silent. If not, the node waits for a random interval and attempts retransmission. We note that packet collisions are known to be rare when employing pulse-based modulation (e.g., TS-OOK), on the condition that inter-pulse periods are picked at random per node pair [15]. In other words, each node pair must negotiate its inter-pulse period, implying applicability to peer-to-peer communications only. On the contrary, the flood-based paradigm yields one-to-many data dissemination, where such negotiations have no point. Thus, the inter-pulse period remains constant over all packet exchanges and collisions become possible [15]. Thus, CSMA/CA is considered, as a simple measure to counter collisions in most modulation cases. Nonetheless, when compared to MG-DI, it adds the complexity that each node should sustain a queue of packets pending for transmission. The simulations assume a queue able to hold 10 packets, which was observed to guard against queue overflows. We also note that the packet-source localization advantage of the classification-based approaches, MG-DI and D-DI, (Fig. 2) cannot be offered by the optimal and CSMA/CA-assisted flood schemes.

**Metrics.** The following metrics are employed to quantify the performance of the compared schemes:

The *network coverage*, defined as the maximum percentage of nodes reached by a new packet emitted from a beacon. More formally, should the beacon emit $b = 1 \ldots B$ packets during a simulation run, the coverage $C$ is defined as:

$$
C = E_{b=1 \ldots B} \left[ \left\| \left\{ n : b \text{ is received} \right\} \right\| / N \right]
$$

where $N$ is the total number of nodes in the topology, $n$ is a given node, $\|\cdot\|$ is the cardinality of the set $\cdot$, and $E_x[\cdot]$ is the mean value of $\cdot$ over parameter $x$.

The *network service time*, $S$, defined as the average time required for relaying a new packet from the beacon to any node. Successful packet deliveries are considered only. Let $S_b$ be the set containing all node service times with regard to packet $b$ emitted from the beacon. Then:

$$
S = \frac{\sum_{b=1}^{B} \sum_{s \in S_b} s}{\sum_{b=1}^{B} \|S_b\|}
$$

Two metrics are used for outlining the energy-efficiency of the compared schemes [49]: i) The *network packet transmission rate*, representing the total number of packets transmitted from all nodes (successfully or not), divided by the duration of the simulation. ii) The *network packet collision rate*, representing the number of packets lost due to collisions over the duration of the simulation.

Finally, the *classification time* is defined as the average time required to classify all nodes into "users" or "retransmitters" (ignoring any equalization
delay). This metric quantifies the setup time imposed by the DI-based schemes, and is applicable to MG-DI and D-DI only.

**Topologies.** The compared approaches are evaluated over a range of 2D and 3D layouts. Apart from standard 2D/3D square grids (e.g., Fig. 1b) and uniform random topologies, we employ the additional layouts depicted in Fig. 5. In the 2D case, a selected layout fills a fixed rectangular area, with dimensions $10 \times 10 \text{mm}$, while in the 3D case the layouts are applied to a $10 \times 10 \times 10 \text{mm}$ volume. The exact node positions are derived from the selected layout type and the total number of nodes to be placed. For each 2D and 3D topology, the total number of nodes is varied in the range $[1000, 8000]$ (in steps of 1000) corresponding to progressively denser topologies.

**Channel Models.** The simulations evaluate the proposed scheme in two different channel models:

The first model uses a full-3D ray tracing approach to deduce the propagation paths, their timing and attenuation [10]. Diffractions, reflections and refractions are considered. The nodes are treated as silicon cubes (conductivity $0 \text{ S/m}$, permittivity $2.4 \text{ F/m}$) with a side of 10 $\mu$m. The space among the nodes is filled with air, roughly corresponding to the structure of metamaterials\(^1\) (e.g., [30]). Furthermore, isotropic antennas are assumed in all 2D and 3D cases.

The second approach employs the statistical model of [12]. The Signal-to-Interference-Plus-Noise-Ratio approach is employed to simulate the packet reception process [51]. The connectivity radius is defined by the Tx Power, the Noise Level, the $\text{SINR}_{\text{thresh}}$, and the path loss, $L$:

$$\frac{P_{\text{TX}}}{\text{Noise} \cdot L(\text{radius})} > \text{SINR}_{\text{thresh}}$$

The employed communication parameter values (Table 1) yield via inequality (5) a maximum radius of 0.95 $\text{mm}$.

Molecular absorption due to the air (absorption coefficient $K$ [12]) and shadow fading ($X$ coefficient in dB [52]) are taken into account in both cases:

\(^1\)We note that the wire-frame-like structure that holds the nanonodes in place is not taken into account by the ray tracing process. The opposite would require FDTD-based simulation techniques instead of ray tracing. However, the high computational complexity of the FDTD techniques did not constitute them a good choice for simulating a nanonetwork with thousands of nodes.
### Table 1: Simulation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Communication Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>100 GHz</td>
</tr>
<tr>
<td>Tx Power (P_{TX})</td>
<td>2 dBnW</td>
</tr>
<tr>
<td>Noise Level</td>
<td>0dBnW</td>
</tr>
<tr>
<td>SINR(_{\text{thresh}})</td>
<td>-10 dB</td>
</tr>
<tr>
<td>Guard Interval</td>
<td>0.1 nsec</td>
</tr>
<tr>
<td>Packet Duration</td>
<td>10 nsec</td>
</tr>
<tr>
<td>Absorption Coefficient (K)</td>
<td>0.52 dB/Km (default)</td>
</tr>
<tr>
<td>Shadow Fading Coefficient (X)</td>
<td>2 dB (default)</td>
</tr>
<tr>
<td><strong>Energy Parameters (Battery-mode only)</strong></td>
<td></td>
</tr>
<tr>
<td>Capacity</td>
<td>1000 (\mu)J</td>
</tr>
<tr>
<td>Energy-Harvesting Rate</td>
<td>variable</td>
</tr>
<tr>
<td>Battery drain per Packet Transmission</td>
<td>100 (\mu)J</td>
</tr>
<tr>
<td><strong>Simulation Run Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Simulation Duration (in packets emitted from beacon)</td>
<td>500</td>
</tr>
<tr>
<td>Equalization Delay (MG-DI)</td>
<td>10 received packets</td>
</tr>
</tbody>
</table>

The default value of \(K\) is set to 0.52 dB/Km, which corresponds to absorption due to standard atmospheric gases at 100 GHz (see [14, p. 3 and p. 16]). Parametric runs with regard to \(K\) are also executed \((K \in [0, 10] \text{ dB/Km})\), in order to deduce the sensitivity of the proposed data dissemination scheme to this parameter in general. Qualitatively, a value of \(K = 0\) corresponds to no absorption-related losses. A value of \(K \approx 0.25\) represents dry air with no water vapor at 100 GHz (0 gr/m\(^3\)). Values of \(K > 0.25\) correspond to increasing presence of water vapor. The default value \(K = 0.52\) corresponds to normal humidity (7.5 gr/m\(^3\)). Thus, the range \(K \in [0.25, 10]\) reflects cases with zero to very high humidity conditions within the nanonetwork.

The shadow fading coefficient is modeled as a Gaussian random variable with standard deviation \(X\) in dB, varying the path loss as \(L_{AB}(\text{radius}) + X\) [52, 53]. To the best of the authors’ knowledge, there are presently no real-world measurements of \(X\) that pertain specifically to nanonetworking environments. On one hand, higher-frequency studies (300 GHz) approximate \(X\) at 1.905 dB for \(\sim\) cm distances [52]. On the other hand, studies at 94 GHz—but at \(\sim\) m distances—place \(X\) within the range 0.6 – 5 dB, with the value 1.8 being representative for most studied cases [53]. In light of these studies, and in absence of nanonetworking-specific measurements, we will use a default value of \(X = 2\) dB and perform parameter variation studies in the range \(X \in [0, 5]\) dB.

**Simulation setup and parameters.** All subsequent runs assume that the node that is nearest to the center of the studied area (rectangle or cube) serves as the beacon, e.g., simulating a source of a sensing event. In any scenario, the beacon emits fixed-sized packets periodically (beacon interval), with the cor-
responding parameters given in Table 1. The choice of operating frequency is inline with Section 2. The Tx Power, Noise Level and SINR threshold parameters are set by expectation [1], since real-world nanonetworks have not yet been manufactured, to the best of the authors knowledge. Notice, however, that their values need only hold by ratio, as expressed by equation (5). The chosen Tx Power (and higher) can be attained by increasing the power of the external coil (see Fig. 1b). The noise level in nanonetworks is expected to be high compared to their Tx Power, due to the presence of molecular noise apart from the common thermal noise [12]. As a result, studies have shown that nanonetworks will be able to operate under very low SINR thresholds [54]. Nonetheless, theoretical lower bounds exist for this parameter. We therefore choose indicatively the mean lower bound for the simplest kind of receiver, i.e., $-10$ dB [55]. A Guard Interval of 0.1 nsec is assumed, meaning that multiple receptions of the same packet arriving within this interval add up to the power of the useful signal. This choice expresses the expectation that the weak nanonode hardware may be sensitive even to mild interference.

The D-DI-specific parameters are taken from [35]. We employ the min, max and mean operators to perform the waveform reductions mentioned in Section 4.1. In absence of a way for automatic reduction type selection, all presented metrics of D-DI refer to optimal values deduced via exhaustive simulations over the three reduction types.

Finally, Figures 6-8 assume inductive power supply, while battery mode operation is evaluated in Figure 9.

**Results.** Figure 6 studies the case of 2000 nodes arranged in different layouts. We employ the statistical channel model and a beacon interval of 1 nsec. According to Fig. 6a, only CSMA/CA and the proposed MG-DI can yield acceptable coverage of 95 − 100%. (This phenomenon is explained later in Fig. 7). Therefore, we restrict the ensuing comparison between CSMA/CA and MG-DI.

In terms of network service time, MG-DI offers the best performance, as shown in Fig. 6b. The ray-like MG-DI classification of the retransmitters (e.g., Fig. 2) has the additional benefit that packets travel in straight paths, rather than random ones. Thus, the network service time is reduced as well. This is shown to also hold for random layouts. A 3D layout imposes a higher service time for all algorithms. This is expected, given that the same number of nodes is now spread over a 3D volume, increasing their distances. We note that network service times are very important for software-defined materials, allowing them to tune their EM behavior in shorter time intervals [4]. Figure 6c shows that the proposed MG-DI can achieve the coverage and service time performance with just half the network packet retransmissions than CSMA/CA, implying increased energy efficiency. This is explained when studying the network packet collision rate in Fig. 6d. MG-DI induces a collision rate approximately 5 times smaller than CSMA/CA, despite the interference-minimizing orientation of the latter. The deterministic approach of MG-DI (i.e., node classification) is shown to outperform the probabilistic nature of CSMA/CA. The 3D layout yields fewer collisions for all algorithms, which is also attributed to the dimensionality increase. Finally, OPTIMAL FLOOD and D-DI offer seemingly better
performance in Fig. 6b-6d, but these results are impractical due to low coverage (Fig. 6a). With less nodes covered, the respective service times, retransmissions and collisions are naturally decreased.

The low performance of OPTIMAL FLOOD and D-DI is explained via Fig. 7, which studies the effects of different channel model configurations. At first, in Fig. 7a, we set $X = 0 \, dB$ in order to study the effects of molecular absorption ($K$ factor). As expected, the effects of molecular absorption is not significant at the studied frequency, as indicated by related studies [14, 12], despite in-
creasing $K$ beyond its natural value. Thus, the reason behind the variation in the performance of the compared algorithms must be sought elsewhere. In Fig. 7b we keep $K$ constant to its natural value and vary the fading coefficient $X$. The effects of fading are significant for all algorithms. At first, the OPTIMAL FLOOD approach is shown to be quite sensitive to fading effects. This outcome is attributed to the minimal-retransmissions objective of this approach. The OPTIMAL FLOOD enforces the smallest packet forwarding rate that is sufficient to cover the complete topology. This may constitute the OPTIMAL FLOOD a good approach in terms of energy efficiency. However, the same criterion also implies that any packet loss is critical, leaving network segments uncovered. Thus, OPTIMAL FLOOD requires adequately good channel conditions to operate, which may not be possible in dense nanonetworks. D-DI is shown to be even more sensitive to fading effects. Notice that D-DI performs the node classification at physical layer, by constantly monitoring power level variations of the incoming signals. Thus, a widely varying signal level (higher $X$ values) forces the D-DI to classify many nodes as users, eventually segmenting the network. On the other hand, MG-DI and CSMA/CA are less sensitive to fading. MG-DI and CSMA/CA operate at a higher layer than D-DI and OPTIMAL FLOOD. MG-DI merely counts packets, while CSMA/CA retransmits when the channel is free, abstracting all other physical-layer details. The packet count is naturally affected more easily than channel availability checks. Thus, CSMA/CA is the least sensitive approach in terms of fading.

The relative ranking of the algorithms does not change when adopting a more realistic, ray-trace-based channel model. However, as shown in both Fig. 7a and 7b, the statistical channel model is more optimistic than the ray-trace-based one in all cases, even up to 40%. Once again, the more physical-layer-aware the approach, the greater the effect on its performance. Thus, D-DI is affected the most by the different channel type. MG-DI is also significantly affected by the channel type at high $X$ values, i.e., when the packet count statistics begin to change drastically as well. Thus, the EM scattering effect of the nanonodes themselves may not be overlooked in dense networks.

Figure 8 studies the performance of the compared schemes when the num-
ber of nodes increases. A 2D Grid layout is employed, noting that the results are similar for the remaining layouts. Since the number of nodes increases and the studied area remains constant, the increase in node numbers corresponds to increased node density. MG-DI and CSMA/CA achieve the best coverage, as shown in Fig. 8a, while the performance of all compared schemes is constant. D-DI exhibits a marginally decreasing coverage, due to the naturally increased interfering signal levels in denser topologies. Once again, physical-layer-aware algorithms exhibit sensitivity to increases in node density. It is also noted that the classification times of MG-DI and D-DI are also constant with regard to node density (Fig. 8b). MG-DI classifies nodes after processing a fixed number of packets (including retransmissions), as explained in Section 4. Thus, the classification time is approximately equal to the beacon interval (1 msec), regardless of density. D-DI classifies nodes much slower (≈ 100 msec), as explained in Section 4.1.

Finally, we study the performance of the compared algorithms in battery mode operation in Figure 9. A 2000-node, 2D Grid layout is assumed. At first, the beacon interval is increased to $\tau = 10$ sec, for the reasons given in Section 4.2. Instead of relying on inductive coupling, each node is equipped with a 1000 $\mu$J battery, while each packet transmission is assumed to consume 100 $\mu$J [1]. Furthermore, a node may harvest energy with a rate of 100 $\mu$J (i.e., 10% of the battery capacity) per a time interval $T$ (sec) which varies during the simulation duration. In this manner, we simulate variations in the harvesting rate, as discussed in related studies [17, 16]. Specifically, $T$ follows the Poisson distribution with a mean value of $T$. Separate runs are executed for $T = n \cdot \tau$, $n \in [1, 20]$, $n \in \mathbb{N}^*$, forming the $x$-axis of Fig. 9. The retransmission probability of user nodes (MG-DI) is set to 15% (cf. Section 4.2). Naturally, a retransmitter-node does not operate when its battery level cannot sustain a packet transmission.

The plot of the achieved coverage versus $T$ is given in Figure 9. At $T = 1 \cdot \tau$ the network can in principle operate perpetually, since the energy drain is equal to the energy harvest rate. The proposed MG-DI and CSMA/CA indeed achieve perpetual operation in practice. On the other hand, D-DI and OPTIMAL FLOOD still suffer from high sensitivity to channel conditions and may not operate satisfactorily. As $T$ increases, the energy reserves of the network are put to strain, due to the low energy harvesting rate. Thus, the performance of all schemes reduces inevitably. It is noted, however, that battery mode operation is more fitting to point-to-point communications [6], whereas the compared schemes are broadcast-oriented. Nonetheless, it is shown that perpetual operation is still possible even in this case, provided that the energy drain is close to the energy harvest rate.

$^2$Notice that values employed in related studies, [1, 16, 17], all correspond by analogy to a battery able to support ~10 consecutive packet transmissions, and to a harvesting unit able to salvage energy for one packet transmission per ~10 sec (cf. Section 2). Any scaled values that respect this analogy produce identical results in the simulations.
In summary, the proposed MG-DI is shown to provide optimal network coverage with fewer resources than the more complex CSMA/CA. Specifically, MG-DI offers smaller network service time, packet transmission and packet collision rates than CSMA/CA (Fig. 6). Furthermore, it is less sensitive to unreliable channel conditions when compared to OPTIMAL FLOOD and D-DI (Fig. 7). It is also noted that the EM scattering effects caused by the nanonodes themselves is significant (Fig. 7b). In addition, the runtime of MG-DI is trivial compared to nature-inspired approaches (Fig. 8). Finally, perpetual operation on battery mode is also supported, despite the intended broadcast mode of operation (Fig. 9).

6. Conclusion

The present paper studied dense, ad hoc nanonetworks with numerous nodes, which are the key-enablers for novel applications of nanonetworking. Introducing cost and scalability considerations, fitting networking approaches should combine extremely simple node architectures with high network coverage and energy efficiency. To this end, the introduced MG-DI scheme proposes the early classification of nanonodes into infrastructure-members and infrastructure-users. The classification is performed using simple packet reception statistics on each node, without need for parametrization or orchestration. Infrastructure nodes then blindly retransmit incoming packets, serving the user-nodes. The classification process can run in non-blocking mode and has trivial complexity. MG-DI was thoroughly evaluated via simulations in various 2D and 3D topologies, taking into account the electromagnetic scattering caused by the nanonodes themselves. The results yielded high network coverage and reduced packet transmission rates with regard to related solutions.

References


