Packet Routing in 3D Nanonetworks: A Lightweight, Linear-path Scheme

A. Tsioliaridou[a], C. Liaskos[a], E. Dedu[b,*], S. Ioannidis[a]

[a] *Foundation of Research and Technology-Hellas, Heraklion, Greece*
[b] *Univ. Bourgogne Franche-Comté / FEMTO-ST Institute/CNRS, Montbéliard, France*

Abstract

Packet routing in nanonetworks requires novel approaches, which can cope with the extreme limitations posed by the nano-scale. Highly lossy wireless channels, extremely limited hardware capabilities and non-unique node identifiers are among the restrictions. The present work offers an addressing and routing solution for static 3D nanonetworks that find applications in material monitoring and programmatic property tuning. The addressing process relies on virtual coordinates from multiple, alternative anchor point sets that act as *viewports*. Each viewport offers different address granularity within the network space, and its selection is optimized by a packet sending node using a novel heuristic. Regarding routing, each node can deduce whether it is located on the linear segment connecting the sender to the recipient node. This deduction is made using integer calculations, node-local information and in a stateless manner, minimizing the computational and storage overhead of the proposed scheme. Most importantly, the nodes can regulate the width of the linear path, thus trading energy efficiency (redundant transmissions) for increased path diversity. This trait can enable future adaptive routing schemes. Extensive evaluation via simulations highlights the advantages of the novel scheme over related approaches.

Keywords: Electromagnetic nano-networking, multi-hop communication, packet routing.

1. Introduction

Nanonetworking is rapidly gaining ground as a key-enabler for novel industrial and medical applications [2]. Mobile nanonetworks are envisioned in collaborating swarms of nano-bots [3] or as highly efficient, programmable drug delivery and virus detection systems within biological organisms. Static nanonetworks find applications in the monitoring of mission-critical industrial materials,

[*] Corresponding author

Email addresses: atsiolia@ics.forth.gr (A. Tsioliaridou), cliaskos@ics.forth.gr (C. Liaskos), eugen.dedu@univ-fcomte.fr (E. Dedu), sotiris@ics.forth.gr (S. Ioannidis)

Preprint submitted to Elsevier Nano Communications Networks *January 10, 2017*

such as nuclear reactor shielding. However, constructing and operating a network comprising numerous, nano-sized nodes creates new technical challenges. This paper focuses on the networking layer and studies the open research issues of nano-node addressing and data routing. The novel solution is applicable to large, static 3D nanonetworks, with omnidirectional antennas and very small node communication radius. Such networks exchange data packets in a massive multi-hop fashion. They find exotic applications within active materials, which can receive external commands and tune their electromagnetic behavior accordingly [17].

Technical and physical limitations at nano-scale call for novel approaches to nano-node addressing and packet routing [27]. Power supply units constitute one of the most critical factors under research. Systems with autonomous nano-nodes must rely on energy scavenging modules, which presently yield enough power for 1 packet transmission per approximately 10 seconds [10]. Wireless power transfer offers a more effective alternative, at the expense of requiring an external power supply [24]. Wireless nano-communication modules, which are expected to operate at the THz band, pose additional limitations, translating to highly lossy channel conditions due to acute molecular absorption phenomena [27]. Finally, manufacturing restrictions and cost scalability considerations correspond to cheap nano-node hardware, i.e., limited CPU capabilities and data storage potential [2]. The impact of these restrictions on nano-node addressing is that assigning unique identifiers to each node is not scalable, mainly due to power restrictions. Regarding data routing, on the other hand, the expectedly frequent transmission failures require a mechanism to balance path redundancy and energy consumption, while incurring low computational complexity and memory overhead [27].

The proposed solution constitutes a routing and addressing scheme that complies with the nano-scale restrictions. The node addressing module relies on virtual coordinates, i.e., geo-locations measured as node distances from a set of anchor points, collectively referred to as viewport. Many alternative viewports are considered, and it is shown that each one offers good address granularity to certain network areas. Thus, the anchor selection depends on the location of the communicating nodes. Subsequently, a viewport selection heuristic is proposed, which enables a sender-node to select the best viewport among available options. Moreover, a stateless routing scheme is proposed, which runs on top of the addressing scheme. According to it, a node can deduce whether it is located on the line connecting the packet sender to the packet destination. Due to its stateless operation, the scheme does not require inter-node clock synchronization or protocol handshake [5]. It offers tunable path redundancy by controlling the width of the 3D line, which constitutes its major advantage over related solutions [23]. Additionally, it requires node-local information and integer computations only, respecting the nano-CPU limitations. Finally, due to its stateless nature, it does not require routing tables for its operation and, hence, no permanent storage as well.

The remainder of this paper is organized as follows. Related studies are given in Section 2. An overview of nanonode addressing is given in Section 3.

Section 4 details the novel concept and introduces the proposed routing scheme. Evaluation via simulations takes place in Section 5. Finally, the conclusion is given in Section 6.

2. Related work

The proposed Stateless Linear-path Routing scheme (SLR) is a joint coordinate and routing system. Therefore, in this section we shortly outline the relevant work in both fields and point out the differentiation points of SLR.

2.1. Localization-Coordinate systems for wireless networks

The localization and ranging of nodes in wireless networks can be set based on signal indication attributes (such as the received signal strength), on the signal time of flight, or on coordinate systems. The SLR localization and addressing approach falls into the coordinate systems category, with numerous related solutions pertaining to wireless networks [6]:

- Polar coordinate systems, extensively used in radar systems, employing a vector length as one coordinate, as in SLR, but an angle as second coordinate. Thus, this coordinate system has only one origin, whereas SLR employs three separate origins.

- Geographic coordinate systems, which are specific to Earth, and allow every location on the Earth to be specified by its latitude, longitude and altitude. Such coordinates are specific to the surface of Earth, and are thusly different than SLR. Moreover, such systems typically employ more than four origins for improved accuracy, and distances towards them are measured via signal time of flight. The location of the origins is fixed and known in advance. Floating point precision is required to convert between origin distances and the location coordinates. In contrast, in SLR distances are expressed as *integer* numbers, and only integer processing is internally used. Moreover, the locations of the the origins remain unknown to network nodes.

- Trilinear coordinates, applied in wireless sensor networks, which use three references: the coordinates are the distances to the three sidelines (i.e. the infinite line that contain the edge) of a triangle. In contrast, SLR uses points as origins.

SLR can be classified as a particular case of curvilinear system. Curvilinear coordinate systems are systems in which the coordinate lines (Ox, Oy and/or Oz) may be curved instead of straight line, cf. Fig. 2. More precisely, SLR is equivalent to the elliptic coordinate system, which is defined as a system in which coordinate lines (i.e. the lines obtained when one of the coordinates is kept constant) are confocal ellipses and hyperbolas. Coordinates (x, y) in

SLR system of origins $A_1(0,0,a)$ and $A_2(0,0,-a)$ can be obtained from elliptic coordinates (u,v) as follows:

$$(x,y) = (a(u+v), a(u-v)) \tag{1}$$

and reciprocally:

$$(u,v) = \left(\frac{x+y}{2a}, \frac{x-y}{2a}\right) \tag{2}$$

2.2. Routing

Related studies on data routing within propagation focused on setups pertaining to Body Area Network (BAN) applications [21]. These networks comprise sparse, full-mesh topologies of mobile nodes. Due to their low numbers, the nodes are commonly assumed to have unique identifiers. Furthermore, BAN-oriented studies assume hierarchical networks, where a set of relatively powerful nano-routers control the smaller, weaker and cheaper nanonodes [20]. In this context, studies have focused on Medium Access Control (MAC) schemes that take into consideration the energy harvesting rate to achieve perpetual operation [1, 19]. Subsequently, energy-aware neighborhood discovery and node handshake processes are proposed in these studies. Due to the presence of the larger nano-routers, the hierarchical approach is too intrusive for the smart material applications targeted by the present study [17]. Additionally, such applications require large networks, and minimal transmission power per node, which translates to multi-hop routing requirements and non-unique addressing.

Due to their mutual emphasis on low-overhead communication, routing in nanonetworks exhibits similarities to wireless sensor networks (WSN) [31]. However, the restrictions of the nano-environment invalidate existing WSN routing solutions. It has been experimentally confirmed that well-known routing protocols, such as AODV, DSDV, or DSR are not scalable in terms of nodes or energy expenditure [15]. More lightweight geo-addressing WSN approaches provide satisfactory performance for large networks [15]. Nonetheless, such routing algorithms require directional routing, attained with neighborhood discovery, routing tables, and even memorization of past node attributes (last-seen location, direction, velocity).

Similarities also exist between routing in nanonetworks and networks-on-chips (NoCs) [30]. NoCs need to discover their topology and perform defect mapping in a completely decentralized manner [4]. Nonetheless, NoCs assume few, powerful nodes, which rely on external power supply and not on scavenging [30]. Therefore, NoC-oriented solutions are generally not portable to nanonetworks.

Regarding the authors' prior work, a ray-tracing-based simulation technique for nanonetworks was studied in [12]. Liaskos et al. proposed a metaheuristic-based, selective flooding dissemination scheme for 2D nanonetworks [16], which was later refined in terms of complexity [24]. Tsioliaridou et al. proposed a joint peer-to-peer routing and geo-addressing approach for 2D networks [23], which was subsequently extended to 3D [25]. However, the routing module in these

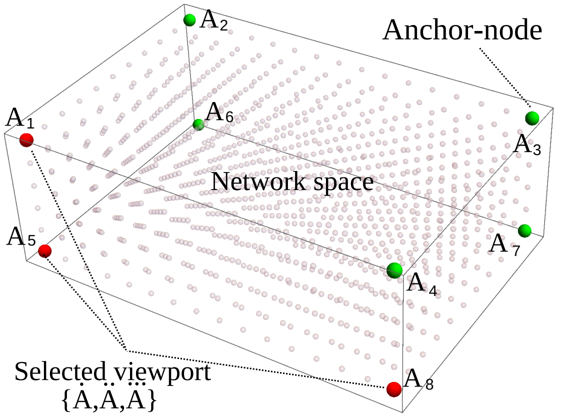


Figure 1: Overview of the studied nanonetwork.

studies operated via selective flooding within network *volumes*. The present paper differentiates by proposing a routing approach over curvilinear paths, accomplished in a stateless manner.

An early version of this work, which does not include the viewport selection optimization process, can be found in [22].

3. Nanonode addressing in 3D

Figure 1 illustrates the considered system setup, comprising a set of nanonodes placed within a 3D rectangular *network space*. The layout of the nodes is either a regular grid or random within the space. The grid setup corresponds to an active meta-material application, while the random placement to a smart material monitoring setup [17]. The network nodes have the same hardware. Protocols using mixed omni- and uni-directional antennas have been proposed [29]. The presented SLR scheme assumes uniformity: each node is equipped with identical omnidirectional antennas. This is in line with its restricted-flood-based operating principle: no next node has to be explicitly chosen during the SLR packet routing process. As such, they have a short wireless connectivity radius imposed by the nature of the application [17], and we assume a multi-hop packet routing case. Additionally, the network conditions are such that node failures are common, accentuating the need for alternative paths (path redundancy). Nanonode communication may temporarily fail due to error-prone hardware, energy shortage or channel conditions [11].

Eight nodes, denoted as *anchors*, are placed at the vertexes of the space during the construction of network. These anchors are indexed as shown in Fig. 1 ($A_{1 \ldots 8}$). Note that the anchors are identical to any other node. Their uniqueness pertains to their role in the node *addressing* phase.

Node addressing. This stage happens once and serves as initialization of the system. It allocates addresses to nanonodes, which are saved for the

setup flag (1 bit)	anchor index (3 bits)	hop count

Table 1: Structure of setup packets.

lifetime of the network. We employ the 3D location of a node as its address using a trilateration process [18]. Initially, the nodes obtain their distances from the anchors as follows. The anchor A_1 initiates the addressing phase by broadcasting a data packet with the structure shown in Table 1.

A *setup flag* set to the value "1" denotes that the packet is exchanged as part of the initialization phase. Additionally, the *hop count* integer field is set to 0. Each non-anchor recipient node increases the hop count field by +1 and memorizes the ensuing value as its distance from anchor $A_{anchor\,index}$. Subsequently, it re-broadcasts the packet. These steps are executed only for the first received packet from a given anchor. Should an anchor-node with index $(anchor\,index) + 1$ receive such a packet (e.g., A_2), it is triggered to generate its own setup packet. A trivial timeout is allowed to ensure that the completion of the previous setup packet has been completed. Thus, once the initialization phase is complete, each node has obtained its address as hop distances, $\{r_i,\, i = 1 \ldots 8\}$, from the anchors $A_{1 \ldots 8}$.

Notice that several neighboring nodes are assigned the same address. In other words, an address refers to an area, denoted as *zone*, rather than a node. The fact that all nodes within a zone share the same geo-address allows for a degree of natural node fail-over. Given the frequent node failures in the nano-environment, geo-address sharing increases the chances that the network will remain connected after each failure.

If for any reason a node does not receive at the right time the setup packet sent by an anchor, it can still receive it via the retransmissions of its neighboring nodes. Thus, the node will be still connected, albeit mapped at slightly different coordinates (off by 1 hop). Having some nodes mapped at slightly different positions could add a few spurious communications, without, however, posing a problem to the protocol.

Generally, a point in 3D space is uniquely identified by its distances from 4 anchor points [18]. However, the employed 3D addressing scheme can conditionally use only 3 anchors for this task. The necessary condition for unique zone identification follows.

Lemma 3.1. *A zone can be uniquely identified by three anchors (*viewport*) distances, $\{\dot{r},\, \ddot{r},\, \dddot{r}\}$, located at the same face of the rectangular network space.*

Proof. Assume a rectangular network space with side lengths $X,\ Y,\ Z$, and a Cartesian system, with $x \in [0,\ X]$, $y \in [0,\ Y]$, $z \in [0,\ Z]$. Let a viewport comprising three anchors located on one face of the space:

$$\dot{A} = \{0,\ 0,\ 0\}\,,\ \ddot{A} = \{X,\ 0,\ 0\}\,,\ \dddot{A} = \{0,\ 0,\ Z\} \tag{3}$$

Notice that, by proper rotation and transfer, these anchors can represent any viewport on the same side of a 3D rectangle. Furthermore, let any point P

within the space have Cartesian coordinates, $\{x,\ y,\ z\}$, and distances from the anchors, $\{\dot{r},\ \ddot{r},\ \dddot{r}\}$. These distances fulfil the following equations:

$$z = \sqrt{\dot{r}^2 - x^2 - y^2} \tag{4}$$

$$z = \sqrt{\ddot{r}^2 - (x - X)^2 - y^2} \tag{5}$$

$$z = Z - \sqrt{\dddot{r}^2 - x^2 - y^2} \tag{6}$$

We will show that each $\{\dot{r},\ \ddot{r},\ \dddot{r}\}$ corresponds to a unique $\{x,\ y,\ z\}$ triplet. From (4) and (5) we easily derive x uniquely as:

$$x = \frac{r^2 - \ddot{r}^2 + X^2}{2 \cdot X} \tag{7}$$

z is also uniquely identified from (4) and (6) as:

$$z = \frac{\dot{r}^2 - \dddot{r}^2 + Z^2}{2 \cdot Z} \tag{8}$$

Finally, from (4) we obtain two candidate y values as:

$$y = \pm\sqrt{\dot{r}^2 - x^2 - z^2} \tag{9}$$

However, our initial assumption is that $y \in [0,\ Y]$. Thus, the negative solution can be rejected, leading to the unique definition of the complete triplet, $\{x,\ y,\ z\}$. $\blacksquare$

The proven property enables the linear routing scheme detailed in the ensuing Section.

4. Stateless Linear Routing

The presented Stateless Linear Routing (SLR) defines a way for routing a packet from any sender node P_1 to any recipient P_2, both identified by their addresses. The sender initially selects one viewport out of $A_{1\ldots8}$ (i.e., anchors $\left\{\dot{A}, \ddot{A}, \dddot{A}\right\}$), which is essentially the coordinate system *(CS)* for the packet route towards node P_2. The considerations on the viewport selection process are discussed in the dedicated subsection below. The corresponding *usable addresses (UA)* of P_1 and P_2 are denoted as $UA_1 : \{\dot{r}_1, \ddot{r}_1, \dddot{r}_1\}$ and $UA_2 : \{\dot{r}_2, \ddot{r}_2, \dddot{r}_2\}$. Then, the sender constructs a packet structured as shown in Table 2, where *setup flag* is 0, *packet id* is a random integer number and *DATA* is the useful load of the packet. Any node P receiving this packet: i) checks if a packet with the same id has been already received, at which case it discards the packet. Elsewhere, the node: ii) memorizes the packet id for a trivial amount of time, in order to avoid route loops. iii) Deduces its usable address, $UA : \{\dot{r}, \ddot{r}, \dddot{r}\}$,

setup flag (1 bit)	packet id (8 bits)	CS (3×3 bits)
UA$_1$ (var)	UA$_2$ (var)	DATA (var)

Table 2: Packet format, as constructed by the sender.

based on the CS field. iv) Re-transmits the packet if it is located on the linear path connecting P_1 and P_2. The latter check is performed as follows.

In order to deduce whether a node P is located on the straight line, $\overline{P_1P_2}$, one should ideally convert the UAs of P, P_1, P_2 to Cartesian coordinates, and then check the compliance to the standard relation:

$$\frac{x-x_1}{x_2-x_1}=\frac{y-y_1}{y_2-y_1}=\frac{z-z_1}{z_2-z_1} \tag{10}$$

However, this approach poses two challenges. First, converting UAs to Cartesian coordinates requires knowledge of the overall network space dimensions and floating point processing capabilities, as shown in equations (7)–(9). Notice that the nodes are only aware of their distances from the network anchors. Deriving the network space dimensions requires extra computations with floating point precision as well.

In order to avoid the complexity challenges, we shall treat the UAs, i.e., point-to-anchor distances $P:\{\dot{r},\ddot{r},\dddot{r}\}$, as a curvilinear coordinate system [26]. Thus, the equation of a line connecting two points $P_1:\{\dot{r}_1,\ddot{r}_1,\dddot{r}_1\}$ and $P_2:\{\dot{r}_2,\ddot{r}_2,\dddot{r}_2\}$ becomes:

$$\frac{\dot{r}-\dot{r}_1}{\dot{r}_2-\dot{r}_1}=\frac{\ddot{r}-\ddot{r}_1}{\ddot{r}_2-\ddot{r}_1}=\frac{\dddot{r}-\dddot{r}_1}{\dddot{r}_2-\dddot{r}_1} \tag{11}$$

This approach also justifies the focus of Lemma 3.1 to three anchors, instead of the four required by formal trilateration processes.

The second challenge for linearity check stems from the fact that the distances $\dot{r},\ddot{r},\dddot{r}$ measure number of hops and, therefore, are integer numbers. Thus, relation (11) will not hold precisely in the general case. To address this issue we rewrite relation (11) as:

$$\begin{cases} \left(\dot{r}-\dot{r}_1\right)\left(\ddot{r}_2-\ddot{r}_1\right)-\left(\ddot{r}-\ddot{r}_1\right)\left(\dot{r}_2-\dot{r}_1\right)=0 \\ \left(\dot{r}-\dot{r}_1\right)\left(\dddot{r}_2-\dddot{r}_1\right)-\left(\dddot{r}-\dddot{r}_1\right)\left(\dot{r}_2-\dot{r}_1\right)=0 \end{cases} \tag{12}$$

Subsequently, we define the quantities:

$$\mathbf{\Delta}^{\mathbf{a}}\left(\dot{r},\ddot{r}\right)=\left(\dot{r}-\dot{r}_1\right)\left(\ddot{r}_2-\ddot{r}_1\right)-\left(\ddot{r}-\ddot{r}_1\right)\left(\dot{r}_2-\dot{r}_1\right) \tag{13}$$

$$\mathbf{\Delta}^{\mathbf{b}}\left(\dot{r},\dddot{r}\right)=\left(\dot{r}-\dot{r}_1\right)\left(\dddot{r}_2-\dddot{r}_1\right)-\left(\dddot{r}-\dddot{r}_1\right)\left(\dot{r}_2-\dot{r}_1\right) \tag{14}$$

Therefore, instead of evaluating direct compliance with (11) or (12), we can simply check whether $\mathbf{\Delta}^{\mathbf{a}}$, $\mathbf{\Delta}^{\mathbf{b}}$ undergo a sign change when altering $\dot{r}$, $\ddot{r}$, $\dddot{r}$. Thus, we define the quantities:

$$\begin{gathered} \mathbf{\Delta}^{\mathbf{a}}_{\dot{\mathbf{r}}}\left(\dot{r},\ddot{r}\right)=\mathbf{\Delta}^{\mathbf{a}}\left(\dot{r}-m,\ddot{r}\right) \\ \mathbf{\Delta}^{\mathbf{a}}_{\ddot{\mathbf{r}}}\left(\dot{r},\ddot{r}\right)=\mathbf{\Delta}^{\mathbf{a}}\left(\dot{r},\ddot{r}-m\right) \\ \mathbf{\Delta}^{\mathbf{a}}_{\dot{\mathbf{r}}\ddot{\mathbf{r}}}\left(\dot{r},\ddot{r}\right)=\mathbf{\Delta}^{\mathbf{a}}\left(\dot{r}-m,\ddot{r}-m\right) \end{gathered} \tag{15}$$

$$\begin{gathered}\mathbf{\Delta}^{\mathbf{b}}_{\dot{\mathbf{r}}}\left(\dot{r}, \dddot{r}\right) = \mathbf{\Delta}^{\mathbf{b}}\left(\dot{r} - m, \dddot{r}\right)\\ \mathbf{\Delta}^{\mathbf{b}}_{\dddot{\mathbf{r}}}\left(\dot{r}, \dddot{r}\right) = \mathbf{\Delta}^{\mathbf{b}}\left(\dot{r}, \dddot{r} - m\right)\\ \mathbf{\Delta}^{\mathbf{b}}_{\dot{\mathbf{r}}\dddot{\mathbf{r}}}\left(\dot{r}, \dddot{r}\right) = \mathbf{\Delta}^{\mathbf{b}}\left(\dot{r} - m, \dddot{r} - m\right)\end{gathered} \tag{16}$$

where m is a non-zero integer. Thus, a node can deduce whether it is placed on a line by performing the following checks:

$$\begin{gathered}\left[\mathbf{\Delta}^{\mathbf{a}} \cdot \mathbf{\Delta}^{\mathbf{a}}_{\dot{\mathbf{r}}} \leq 0 \quad \text{OR} \quad \mathbf{\Delta}^{\mathbf{a}} \cdot \mathbf{\Delta}^{\mathbf{a}}_{\ddot{\mathbf{r}}} \leq 0 \quad \text{OR} \quad \mathbf{\Delta}^{\mathbf{a}} \cdot \mathbf{\Delta}^{\mathbf{a}}_{\dot{\mathbf{r}}\ddot{\mathbf{r}}} \leq 0\right]\\ \text{AND}\\ \left[\mathbf{\Delta}^{\mathbf{b}} \cdot \mathbf{\Delta}^{\mathbf{b}}_{\dot{\mathbf{r}}} \leq 0 \quad \text{OR} \quad \mathbf{\Delta}^{\mathbf{b}} \cdot \mathbf{\Delta}^{\mathbf{b}}_{\dddot{\mathbf{r}}} \leq 0 \quad \text{OR} \quad \mathbf{\Delta}^{\mathbf{b}} \cdot \mathbf{\Delta}^{\mathbf{b}}_{\dot{\mathbf{r}}\dddot{\mathbf{r}}} \leq 0\right]\end{gathered} \tag{17}$$

The integer m controls the "width" of the linear path, essentially introducing a way of meeting the requirement for path redundancy in nanonetworking. A value of $m = 1$ corresponds to minimal redundancy, while greater values increase the number of alternative paths. The value of m could be a global preset, or a part of the packet header set by the original sender. For example, a sender may initially choose a value of $m = 1$. If no delivery acknowledgment is received, the sender can then retry with a value of $m = 2$, etc. The redundancy introduced by m is complementary to the zone redundancy, enforced by the addressing scheme as discussed in Section 3. Zone redundancy favors network connectivity, while m introduces path redundancy by employing additional zones.

The condition (17) checks whether a node is located on a linear path of width m, defined by P_1 and P_2. Additionally, a node can check whether it is on the line *segment* connecting P_1 to P_2 based on the condition:

$$\left(\dot{r} - \dot{r}_2\right)\left(\dot{r} - \dot{r}_1\right) \leq 0 \quad \text{AND} \quad \left(\ddot{r} - \ddot{r}_2\right)\left(\ddot{r} - \ddot{r}_1\right) \leq 0 \quad \text{AND} \\ \left(\dddot{r} - \dddot{r}_2\right)\left(\dddot{r} - \dddot{r}_1\right) \leq 0 \tag{18}$$

Therefore, the linear routing process, outlined as Algorithm 1, consists of checking compliance with condition (18) first, and subsequently with condition (17). If any condition yields false, the packet is consumed and is not retransmitted.

```
Input: An incoming packet pkt at a node $P$.
Output: Retransmission decision.
Derive $UA : \{\dot{r}, \ddot{r}, \dddot{r}\}$ of $P$ from pkt.CS
$UA_1 \leftarrow$ pkt.UA$_1$, $UA_2 \leftarrow$ pkt.UA$_2$
$m \leftarrow$ pkt.m                // in case m is a header of pkt
if Condition(15) AND Condition(16) then
    retransmit pkt
end
```

Algorithm 1: The Stateless Linear Routing process.

Note that the described packet routing process was designed to meet the nanonetworking specifications outlined in Section 1. It inherently provides tunable path redundancy via the m parameter, which also affects the involved number of retransmitters and, hence, the expended energy. Additionally, its

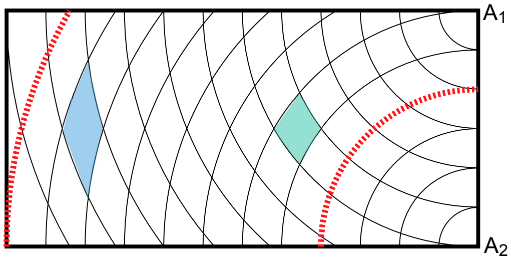


Figure 2: The path curvature and zone resolution trade-off stemming from the selection of anchors.

stateless nature (absence of routing tables) translates to minimal memory overhead. Moreover, it considers the limited nano-CPU capabilities, by requiring integer calculations only.

4.1. Viewport optimization

A consideration stemming from the viewport selection process is illustrated in the 2D example of Fig. 2. The direct mapping of curvilinear coordinates to Cartesian naturally yields curved paths (dashed lines). Should we select anchors A_1, A_2 too close to a communicating pair the curvature is high, while distant anchors yield approximately straight paths. On the other hand, the zone resolution of the addressing system is better (i.e., fewer nodes per zone) for medium pair-to-anchors distances (shaded areas, Fig. 2). Ideally, the path should be straight, with high zone-granularity. We proceed to study this aspect in the current subsection.

According to the described routing process, the packet origin node, P_1, is given the option to select the optimal viewport (coordinate system comprising an anchor triplet) that will handle the complete packet propagation to P_2. According to Lemma 3.1, there exist 24 valid viewport options (4 on each face of the space), denoted as the set $\mathcal{CS}$. Furthermore, each selection affects the involved number of retransmitters, as shown qualitatively in Fig. 2. We will assume that the number of retransmitters per volume unit is constant within the network space. Therefore, we seek the viewport, $cs_o \in \mathcal{CS}$, that minimizes the volume defined by condition (17) `AND` (18) as follows:

$$cs_o = \text{argmin}_{(cs \in \mathcal{CS})} \left\{ \int_{x=0}^{X} \int_{y=0}^{Y} \int_{z=0}^{Z} Predicate\left((17)\ \texttt{AND}\ (18)\right) dx\, dy\, dz \right\} \tag{19}$$

where $Predicate(*)$ is a function that returns 1 when condition $*$ is true, and 0 otherwise. Additionally, $\{x, y, z\}$ are the Cartesian coordinates corresponding to a triplet $\{\dot{r}, \ddot{r}, \dddot{r}\}$ via equations (7), (8) and (9), while the triplet $\{\dot{r}, \ddot{r}, \dddot{r}\}$ itself is a function on the chosen viewport, cs.

The exact calculation of equation (19) is prohibitive for a hardware-constrained nano-node, which is also assumed capable for integer-processing only in our case.

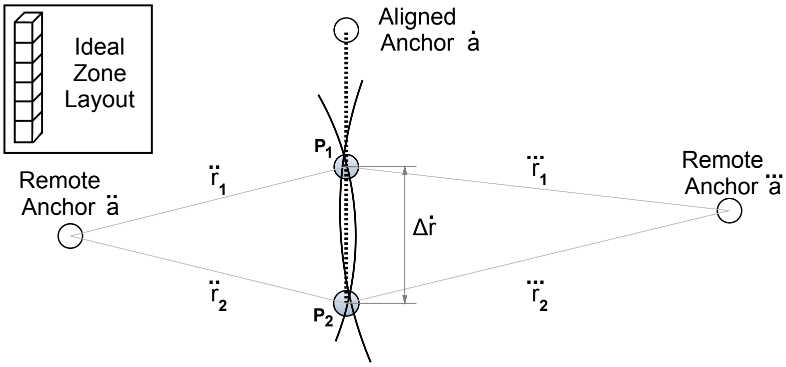


Figure 3: The model-viewpoint for a communicating node pair P_1, P_2.

Thus, we propose an alternative, heuristic approach that complies with these computational restrictions. The heuristic logic relies on model-fitting: instead of solving (19) to obtain cs_o, we search for the cs that most closely matches a good model-viewpoint.

The rationale for the employed model-viewpoint is illustrated in Fig. 3. The model comprises three anchors, in accordance with Lemma 3.1, which are denoted as one *aligned anchor* ȧ, and two *remote anchors* ä, $\dddot{\text{a}}$. The aligned anchor is located on the line defined by nodes P_1 and P_2. In this manner, a maximal zone resolution over the straight path connecting P_1 to P_2 is ensured, as per the ideal layout shown in Fig. 3. In contrast, an anchor, e.g., perpendicular to the segment $\overline{P_1P_2}$ would see limited or no zone differentiation over it. Remote anchors ä, $\dddot{\text{a}}$ are located as far as possible from nodes P_1 and P_2. Ideally, this translates to negligible curvature over nodes P_1 and P_2, producing an almost straight path. The aligned and remote anchors combined yield the ideal zone layout of Fig. 3.

Finding the anchor that best matches the aligned anchor is straightforward. Essentially, we seek the anchor for which the quantity $\Delta\dot{r}$ in Fig. 3 is maximized. Let A_i, $i = 1 \ldots 8$ be the set of all anchors of the network space. Moreover, let $R(n_1, n_2)$ be the distance between two nodes n_1 and n_2. Then, the aligned anchor ȧ is best approximated as:

$$\dot{\text{a}} \leftarrow argmax_{(A_i)} \{|R(P_1, A_i) - R(P_2, A_i)|\} \tag{20}$$

Finding the farthest anchor from P_1 *and* P_2 (either ä or $\dddot{\text{a}}$) is accomplished as follows:

$$\ddot{\text{a}} \leftarrow argmax_{(A_i)} \{\min\{R(P_1, A_i), R(P_2, A_i)\}\} \tag{21}$$

In other words, the distance of an anchor from a *pair* of nodes P_1, P_2 as a whole, is defined as their minimum distance from the anchor. Notice that both relations (20) and (21) uphold the integer computations-only restriction.

Having defined the aligned and remote anchor approximations, we proceed to define a procedure to obtain the viewpoint that best fits the model of Fig. 3. The process requires a prioritization of either the aligned or the remote anchor definition. For instance, selecting the aligned anchor ȧ first via relation (20) means that the possible viewport choices will now be limited to those containing the chosen ȧ only (i.e., 9 viewports out of 24). A similar narrowing of choices occurs with the prioritization of ä. The prioritization essentially refers to either good zone resolution (by choosing ȧ), or a shorter path between P_1 and P_2 but with larger zones. With no loss of generality, we prioritize good zone resolution since, as shown in Fig. 2, zones can be enlarged and deformed significantly at great distances from the anchors. Thus, the achieved zone layout would differ significantly from the ideal one given in Fig. 3.

The viewport selection process, run by the sender P_1 when preparing to send an original packet, is formulated as Algorithm 2. At step 1, the process selects the aligned anchor in accordance with relation (20). Steps 2, 3 initialize two helper variables that will be utilized to update and eventually hold the best viewport. Steps 4–11 focus on all viewport choices that contain the selected aligned anchor. Each candidate viewport *cs* is broken down to its contents in step 5, while the ordering of ä and $\ddddot{\text{a}}$ is arbitrary. In order to reduce complexity by avoiding a double sorting by ä → $(P_1 P_2)$ and $\ddddot{\text{a}}$ → $(P_1 P_2)$ distances (relation (21)), step 6 calculates a collective distance of anchor-pair to node-pair, $(\ddot{\text{a}}, \ddddot{\text{a}}) \rightarrow (P_1 P_2)$, using the *min*() reduction as well. Steps 7–10 update the helper variables, holding the *cs* that contains the most aligned *and* the most remote anchors. Finally, the usable addresses of P_1 and P_2 are set at steps 12–13 and are incorporated to the packet format of Table 2.

Input: The distances of nodes P_1, P_2 from the anchors $A_{1\ldots8}$.
Output: The usable addresses UA_1 UA_2 of P_1, P_2.

```
1  $\dot{\text{a}} \leftarrow argmax_{(A_i)}\left\{\left|R\left(P_1, A_i\right) - R\left(P_2, A_i\right)\right|\right\}$
2  $best\_cs \leftarrow \emptyset$
3  $max\_distance \leftarrow -1$
4  for $cs \in \mathcal{CS} : \dot{a} \in cs$ do
5      $\{\ddot{\text{a}}, \ddddot{\text{a}}\} \leftarrow cs - \dot{\text{a}}$
6      $D \leftarrow \min\left\{R\left(P_1, \ddot{\text{a}}\right), R\left(P_2, \ddot{\text{a}}\right), R\left(P_1, \ddddot{\text{a}}\right), R\left(P_2, \ddddot{\text{a}}\right)\right\}$
7      if $D > max\_distance$ then
8          $best\_cs \leftarrow cs$
9          $max\_distance \leftarrow D$
10     end
11 end
12 $UA_1 \leftarrow P_1(best\_cs)$
13 $UA_2 \leftarrow P_2(best\_cs)$
```

Algorithm 2: The viewport selection process.

Regarding its complexity, Algorithm 2 requires:

- 16 operations for step 1 (2 operations for each of the 8 anchors),

- 2 operations for steps 2–3 (collectively),
- 81 operations for the for loop (9 operations for each of the 9 iterations),
- 2 operations for steps 12–13 (collectively),

leading to a total of 101 integer calculations. Its memory requirements are 6 integer slots required to hold the variables $\dot{a}$, $\ddot{a}$, $\ddddot{a}$, $best\_cs$, $max\_distance$, D.

5. Simulations

This Section evaluates the routing efficiency of the proposed Stateless Linear Routing (SLR), compared to the CORONA routing approach [23]. CORONA is a recently proposed, stateless and provenly scalable nano-routing approach. It proposes a packet flood approach, contained within the area defined by the minimal and maximal anchor distances of the communicating node-pair. Thus, although originally proposed for 2D networks, CORONA has easily been extended for our purposes to the 3D case as well. The simulator is implemented on the AnyLogic platform [28].

Given that CORONA does not include a viewpoint optimization approach, the evaluation is broken into two parts. Firstly, we evaluate the proposed, model-based viewpoint selection for SLR only (subsection 5.2). Algorithm 2 is compared to: i) the optimal viewpoint selection derived via brute force, and ii) random viewpoint selection. Versions prioritizing zone resolution and anchor distance are studied. The metric of viewpoint selection performance is the number of involved retransmitters per communicating node-pair (fewer is better). Secondly, we evaluate the communication efficiency of SLR versus CORONA (subsection 5.3), keeping a static viewpoint for the duration of the evaluation. The main comparison metric is the communication success probability between any node pair, under the conditions of: i) failures of intermediate nodes due to energy depletion, and ii) simultaneous, potentially interfering communications taking place in parallel.

5.1. Configuration

The employed simulation configuration is given in Table 3 [24]. We assume a number of $N = 5000$ nodes within a cubic 3D network space with side size 1 cm. The nodes are placed according to two layouts: i) on a regular grid with $^{1}/_{16}$ cm spacing, or ii) randomly within the space. The latter case is derived from the regular grid by deactivating a random subset of the nodes, as discussed below (run setup).

Each node represents a nano-controller in a smart meta-material application [17]. The space among the nodes is filled with air composed of standard atmospheric gases at normal humidity [7]. This dense network setup as a whole is intended to approximate the building block for a visible light-interacting smart meta-material [17].

Regarding the communication model, all nodes are equipped with isotropic antennas. Molecular absorption due to the air (absorption coefficient K [9]) and

Parameter	Value
Communication & Power Parameters	
Frequency	100 GHz
Transmission Power	5 dBnW
Noise Level	0 dBnW
Reception SINR threshold	-10 dB
Guard Interval	0.1 nsec
Packet Duration	10 nsec
Path Attenuation Parameters	
Absorption Coefficient K	0.52 dB/Km
Shadow Fading Coefficient X	0.5 dB

Space setup (Number of nodes: 5000**)**	
Cubic $1 \times 1 \times 1$ cm, Grid/Random node placement	

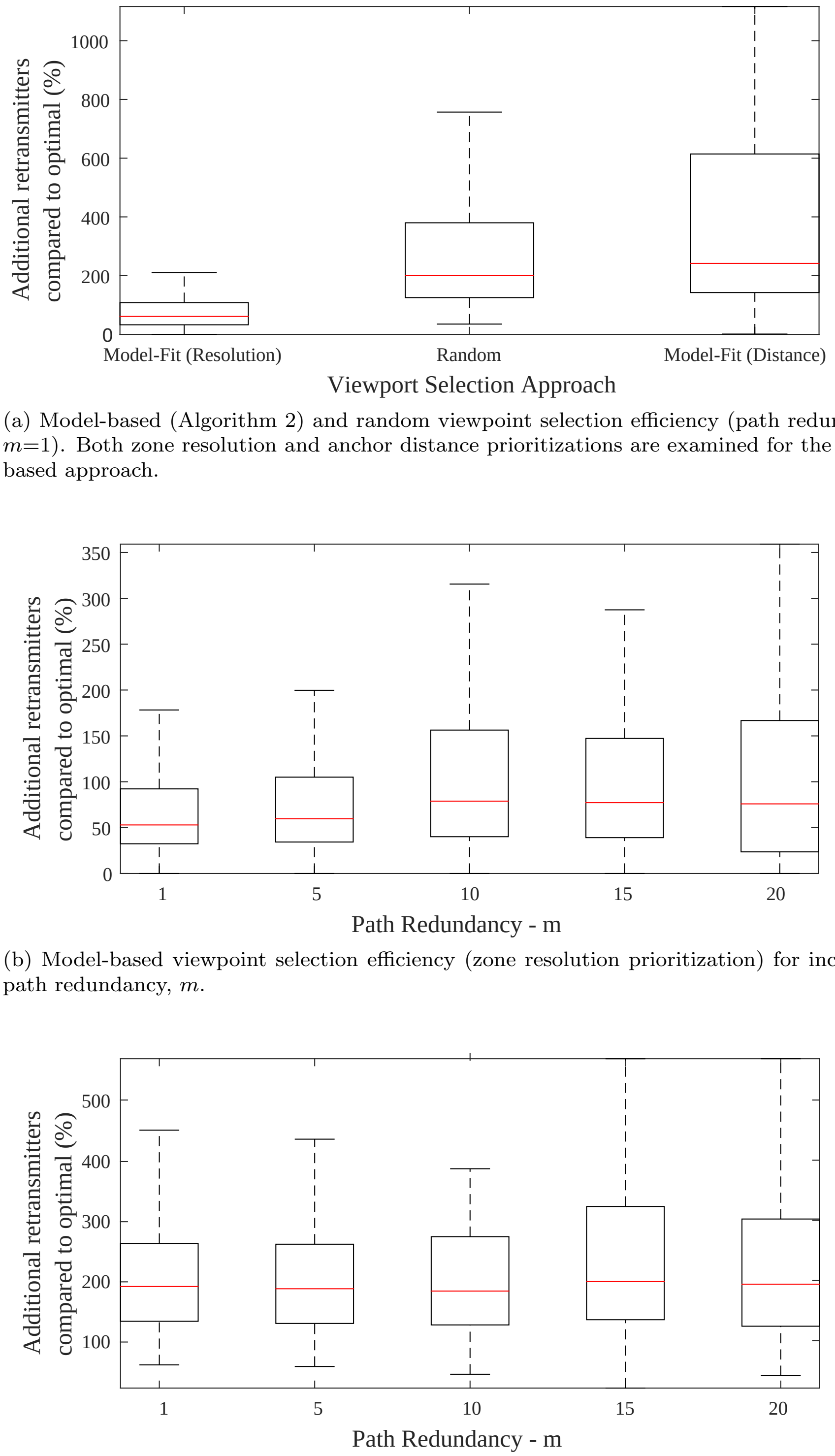
Table 3: Simulation Configuration.

shadow fading (X coefficient in dB [13]) are taken into account. The Signal to Interference plus Noise (SINR) model is used to deduce the success of a packet reception [8, 14]. The transmission power was chosen to ensure that each zone comprises approximately 15 nodes. Thus, the resolution of the network space is $7 \times 7 \times 7$ zones within a $1\,\mathrm{cm}^3$ volume. This low resolution constitutes a *worst-case* scenario for the proposed scheme, since it hinders the formation of well-defined linear paths. The behavior in higher resolution cases is also studied at the end of this Section.

5.2. Evaluation of the viewpoint selection process

Setup. The simulation runs are executed as follows. Once the nodes are placed in their 3D layout (grid or random), we proceed to randomly select 100 node pairs in a sequential manner. For each pair, we execute a packet exchange and log the number of retransmitters yielded by the proposed viewpoint selection approach presented in Section 4.1. Additionally, we study the effects of prioritization (zone *resolution* or anchor *distance*) by swapping lines 1 and 6 in Algorithm 2. Moreover, we exhaustively check all valid coordinate systems (24) per pair, and deduce the one yielding the lowest number of transmitters in each case (*optimal*). The average number of retransmitters over all possible viewpoint choices represents the *random* viewport selection approach. For clarity, all results are relative to the optimal performance. All nodes are considered powered-on in this experiment. The results over all pairs are gathered in the from of boxplots, presented in Fig. 4. The results are similar for both random and grid node layouts.

Results. Initially, in Fig. 4a, it is observed that the model-based selection approach with resolution prioritization outperforms all alternatives. The *resolution* approach yields 50% more retransmitters than the optimal case on average,



(a) Model-based (Algorithm 2) and random viewpoint selection efficiency (path redundancy, m=1). Both zone resolution and anchor distance prioritizations are examined for the model-based approach.

(b) Model-based viewpoint selection efficiency (zone resolution prioritization) for increasing path redundancy, m.

(c) Random viewpoint selection efficiency for increasing path redundancy, m.

Figure 4: Evaluation of viewpoint selection approaches.

which is encouraging for a lightweight scheme that is restricted to employ integer processing only. Note that random selection yields 200% more retransmitters than the optimal case, which is indicative of the gains achieved by the proposed approach. Moreover, the prioritization of zone resolution is shown to be superior to anchor distance, as intuitively expected in Section 4.1, i.e., the distance prioritization produces large zones, leading to a performance similar or worse than the random selection.

We proceed to study the behavior of model-based (resolution prioritization) and random viewport selection as the path redundancy parameter increases. The rationale for this check is that linear paths get thicker as m increases, meaning that the effects of improper viewpoint selection could be magnified. Model-fitting with distance prioritization is not considered due to the preceding results of Fig. 4a. As it can be seen in Fig. 4b (model-based with zone resolution) and Fig. 4c (random), both viewpoint selection approaches yield a good degree of independence from the path redundancy parameter m. The model-based approach retains a performance of 50–75% additional retransmitters than the optimal case, while the random selection is nearly flat around 200%. This phenomenon has a natural explanation. The m parameter is itself a magnifying factor of the volume of retransmitters. The greater the m values, the (proportionately) thicker the linear path. However, a viewport represents an orthogonal concern, i.e., the curvature of the linear path, regardless of its thickness. Thus, the viewpoint selection yields a good degree of independence from the m parameter.

5.3. Evaluation of the networking efficiency

Setup. A run is initiated by forming the described grid arrangement of nodes. A random percentage of nodes (denoted as *deactivation ratio*) is deactivated, emulating failing nodes. Thus, non-zero values of the deactivation ratio have the indirect effect of randomizing the topology as well. Then, a series of 100 operation cycles takes place, with an interarrival of 10 seconds as follows. A number of nodes pairs (denoted as *pair number*) are randomly selected, each requiring the exchange of a single, unique packet. At the end of the 100 cycles, we log the percentage of pairs that communicated successfully (*comm. success ratio*). Each run is repeated 100 times, randomizing the node failures anew, to improve the confidence of the presented results in various topologies.

Results. Figures 5 and 6 illustrate the tunability of the proposed scheme (SLR) versus the related CORONA approach. The pair number is kept constant to 5, and the deactivation ratio varies from 0 to 90%. Figure 5 then shows the attained communication success ratio for various level of SLR path redundancy (m), while Figure 6 presents the corresponding average percentage of the total network nodes serving as intermediate retransmitters per communicating pair. A higher number of retransmitters is evidence of higher energy expenditure rate. The plots in both Figures naturally decrease as the deactivation ratio increases. High deactivation rates yield segmented network areas, reducing the success ratio. Additionally, fewer nodes remain available to serve as retransmitters.

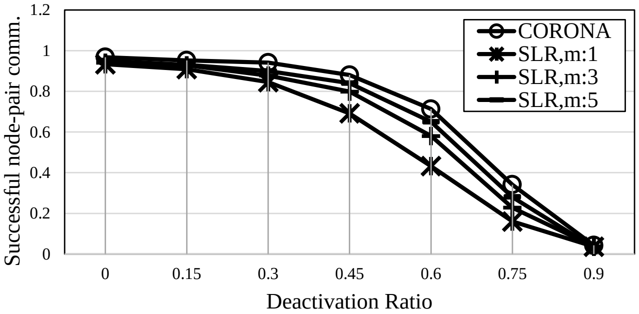


Figure 5: SLR tunability effects on the node-pair communication ratio.

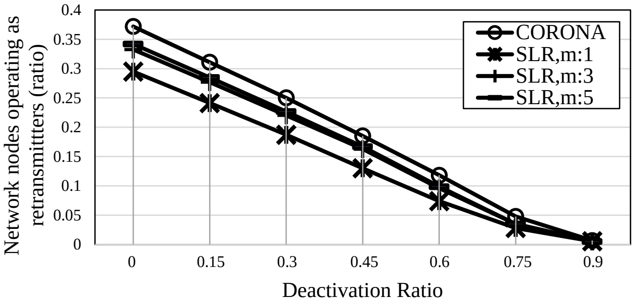


Figure 6: SLR tunability effects on the average, network-wide ratio of retransmitters serving each communicating pair.

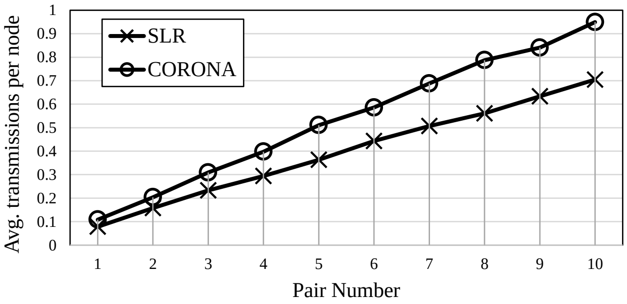


Figure 7: Average packet transmissions per node imposed by CORONA and the proposed SLR, versus the number of concurrently communicating node pairs in the network.

Notably, SLR introduces tunable success ratio and energy efficiency via the path redundancy parameter m, as described in the analysis of Section 4. In other words, SLR allows for a sender node to regulate the network energy expenditure rate, depending on its estimation for the network state. Thus, future schemes can exploit SLR to automatically employ less path redundancy (and, thus, less expense energy) when the network conditions are estimated as relatively good. On the opposite case, when the network conditions are characterized as challenging, the sender node could automatically increase m, attaining higher path diversity and communication success ratio. Mechanisms for network state estimation and automatic adaptation constitute possible extensions of the present work.

We proceed to evaluate the potential for parallel communications offered by SLR in Fig. 7. Keeping the path redundancy and deactivation ratio constant ($m = 1$ and 0% respectively), the pair number is varied in the range 1 to 10, i.e., one to ten node pairs communicating in parallel (x-axis). For each case, we measure the average number of retransmissions imposed to each network node. A higher number of retransmissions implies that the network nodes will deplete their energy reserved faster, thus yielding lower potential for parallel communications. SLR provides better performance over CORONA from this point of view, which is attributed to the well-defined, linear form of the SLR paths. On the other hand, CORONA defines much larger volumes within which packet flooding is executed. Thus, it inevitably incurs more redundant transmissions per node, yielding decreased parallel communications potential.

The presented results are especially promising, given that they refer to a very small space ($1\,\text{cm}^3$), where the gain margin is very limited due to the low zone resolution (cf. “Configuration” above). In Fig. 8 we proceed to study the expected gains in a bigger network space with dimensions $50 \times 50 \times 50\,\text{cm}$. The number of nodes increases proportionally and yields simulation runtime issues, therefore we restrict Fig. 8 to a single, indicative node-pair communication case ($m = 1$). As illustrated, the gains of SLR over CORONA increase considerably in bigger spaces. SLR yields a well-defined arc (i.e., curvilinear path). On the other hand, CORONA produces a much larger volume of restransmitters,

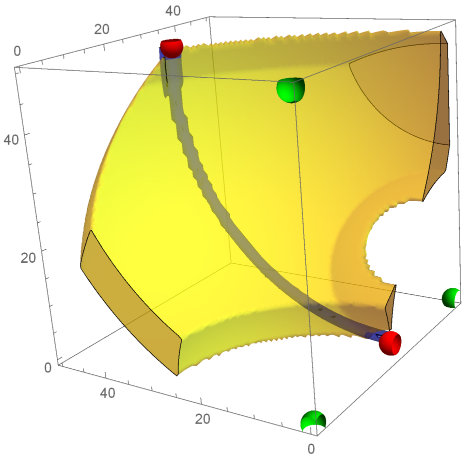


Figure 8: Comparison of CORONA (yellow area) and the novel SLR (dark curve) routing behavior for a given communicating node pair in a *high-resolution* space.

defined by the distances of the communicating nodes from the selected anchors. Thus, CORONA trades routing area size for wider network reachability (i.e., high path redundancy), albeit without tunability potential. SLR, on the other hand, can yield very narrow, linear paths to very wide network areas, setting the basis for adaptivity in the highly challenging nano-environment.

6. Conclusion and future work

The present study introduced a novel addressing and routing scheme for 3D electromagnetic nanonetworks. The scheme allows for tunable routing path redundancy, in order to counter the highly lossy nature of nano-communications while limiting redundant transmissions. Additionally, it yields well-defined, linear routing paths among communicating pairs, allowing for a considerable degree of parallel transmissions within the network. The scheme is stateless, requiring no permanent memory overhead or neighborhood discovery at the nanonode-side. Limitations of nano-CPUs are also taken into account, and each routing decision requires few integer calculations only. The traits of the novel scheme were evaluated via extensive simulations.

Future work includes devising methods to estimate network state in order to adapt line width m automatically, and compose multiple linear path segments to reach non-line-of-sight targets.

Acknowledgement

This work was funded by the European Union via the Horizon 2020: Future Emerging Topics call (FETOPEN), grant EU736876, project VISORSURF

(http://www.visorsurf.eu).

References

[1] Afsharinejad, A., Davy, A., Jennings, B., 2015. Dynamic channel allocation in electromagnetic nanonetworks for high resolution monitoring of plants. Nano Communication Networks 7, 2–16.

[2] Akyildiz, I., Jornet, J., 2010. The Internet of nano-things. IEEE Wireless Communications 17 (6), 58–63.

[3] Boillot, N., Dhoutaut, D., Bourgeois, J., 2015. Large scale mems robots cooperative map building based on realistic simulation of nano-wireless communications. Nano Communication Networks 6 (2), 51–73.

[4] Catania, V., Mineo, A., Monteleone, S., Patti, D., 2014. Distributed topology discovery in self-assembled nano network-on-chip. Computers & Electrical Engineering 40 (8), 292–306.

[5] D'Oro, S., Galluccio, L., Morabito, G., Palazzo, S., 2015. A timing channel-based mac protocol for energy-efficient nanonetworks. Nano Communication Networks 6 (2), 39–50.

[6] Guvenc, I., Chong, C.-C., 2009. A survey on TOA based wireless localization and NLOS mitigation techniques. IEEE Communications Surveys & Tutorials 11 (3), 107–124.

[7] ITU-R, Feb. 2007. Recommendation P.676-7: Attenuation by atmospheric gases.

[8] Iyer, A., Rosenberg, C., Karnik, A., 2009. What is the right model for wireless channel interference? IEEE Transactions on Wireless Communications 8 (5), 2662–2671.

[9] Jornet, J., Akyildiz, I., 2011. Channel Modeling and Capacity Analysis for Electromagnetic Wireless Nanonetworks in the Terahertz Band. IEEE Transactions on Wireless Communications 10 (10), 3211–3221.

[10] Jornet, J., Akyildiz, I., 2012. Joint Energy Harvesting and Communication Analysis for Perpetual Wireless Nanosensor Networks in the Terahertz Band. IEEE Transactions on Nanotechnology 11 (3), 570–580.

[11] Jornet, J., Capdevila, J., Pareta, J., 2012. PHLAME: A Physical Layer Aware MAC protocol for Electromagnetic nanonetworks in the Terahertz Band. Nano Communication Networks 3 (1), 74–81.

[12] Kantelis, K., Amanatiadis, S., Liaskos, C., Kantartzis, N., Konofaos, N., Nicopolitidis, P., Papadimitriou, G., 2014. On the Use of FDTD and Ray-Tracing Schemes in the Nanonetwork Environment. IEEE Communications Letters 18 (10), 1823–1826.

[13] Kim, S., Zajic, A., 2014. A path loss model for 300-GHz wireless channels. In: 2014 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting. pp. 1175–1176.

[14] Lehtomaki, J., Bicen, A., Akyildiz, I., 2014. On the Nanoscale Electromechanical Wireless Communication in the VHF Band. IEEE Transactions on Communications, 1.

[15] Lemmon, C., Lui, S. M., Lee, I., 2009. Geographic forwarding and routing for ad-hoc wireless network: A survey. In: INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on. IEEE, pp. 188–195.

[16] Liaskos, C., Tsioliaridou, A., 2015. A Promise of Realizable, Ultra-Scalable Communications at nano-Scale: A multi-Modal nano-Machine Architecture. IEEE Transactions on Computers 64 (5), 1282–1295.

[17] Liaskos, C., Tsioliaridou, A., Pitsillides, A., Akyildiz, I., Kantartzis, N., Lalas, A., Dimitropoulos, X., Ioannidis, S., Kafesaki, M., Soukoulis, C., 2015. Building Software Defined Materials with Nanonetworks. IEEE Circuits and Systems Magazine 15 (4), 12–25.

[18] Manolakis, D. E., 1996. Efficient solution and performance analysis of 3-d position estimation by trilateration. Aerospace and Electronic Systems, IEEE Transactions on 32 (4), 1239–1248.

[19] Mohrehkesh, S., Weigle, M. C., Das, S. K., 2015. Drih-mac: A distributed receiver-initiated harvesting-aware mac for nanonetworks. Molecular, Biological and Multi-Scale Communications, IEEE Transactions on 1 (1), 97–110.

[20] Pierobon, M., Jornet, J. M., Akkari, N., Almasri, S., Akyildiz, I. F., 2014. A routing framework for energy harvesting wireless nanosensor networks in the Terahertz Band. Wireless Networks 20 (5), 1169–1183.

[21] Piro, G., Boggia, G., Grieco, L., 2015. On the design of an energy-harvesting protocol stack for Body Area Nano-NETworks. Nano Communication Networks 6 (2), 74–84.

[22] Tsioliaridou, A., Liaskos, C., Dedu, E., Ioannidis, S., 2016. Stateless linear-path routing for 3d nanonetworks. In: ACM NANOCOM'16. pp. 28:1–28:6.

[23] Tsioliaridou, A., Liaskos, C., Ioannidis, S., Pitsillides, A., Sep. 2015. CORONA: A Coordinate and Routing system for Nanonetworks. In: ACM International Conference on Nanoscale Computing and Communication. 2. ACM, Boston, Massachusetts, USA, pp. 18:1–18:6.

[24] Tsioliaridou, A., Liaskos, C., Ioannidis, S., Pitsillides, A., 2015. Lightweight, self-tuning data dissemination for dense nanonetworks. Nano Communication Networks, Special Issue: Electromagnetic Communication in Nano-scale 8, 2–15.

[25] Tsioliaridou, A., Liaskos, C., Pachis, L., Ioannidis, S., Pitsillides, A., 2016. N3: Addressing and routing in 3d nanonetworks. In: IEEE ICT'16. pp. 1–6.

[26] Vinson, J. R., 1993. Curvilinear coordinate systems. In: The Behavior of Shells Composed of Isotropic and Composite Materials. Springer, pp. 1–14.

[27] Wang, P., Jornet, J. M., Abbas Malik, M., Fadel, E., Akyildiz, I. F., 2013. Energy and spectrum-aware MAC protocol for perpetual wireless nanosensor networks in the THz Band. Ad Hoc Networks 11 (8), 2541–2555.

[28] XJ Technologies, 2013. The AnyLogic Simulator.
URL `http://www.xjtek.com/anylogic/`

[29] Yao, X.-W., Jornet, J. M., Sep. 2016. TAB-MAC: Assisted beamforming MAC protocol for Terahertz communication networks. Nano Communication Networks 9, 36–42.

[30] Younis, M., Senturk, I., Akkaya, K., Lee, S., Senel, F., 2014. Topology management techniques for tolerating node failures in wireless sensor networks: A survey. Computer Networks 58, 254–283.

[31] Yu, H., Ng, B., Seah, W. K. G., 2015. Forwarding schemes for em-based wireless nanosensor networks in the terahertz band. In: Proceedings of the Second Annual International Conference on Nanoscale Computing and Communication. NANOCOM' 15. ACM, New York, NY, USA, pp. 17:1–17:6.