RIH-MAC: Receiver-initiated harvesting-aware mac for nanonetworks

Article · May 2014

DOI: 10.1145/2619955.2619962

2 authors:

Shahram Mohrehkesh
Temple University
23 PUBLICATIONS **194** CITATIONS
SEE PROFILE

Michele Weigle
Old Dominion University
155 PUBLICATIONS **2,845** CITATIONS
SEE PROFILE

RIH-MAC: Receiver-Initiated Harvesting-aware MAC for NanoNetworks

Shahram Mohrehkesh and Michele C. Weigle
Department of Computer Science, Old Dominion University
{smohrehk, mweigle}@cs.odu.edu

ABSTRACT

In this paper, we introduce RIH-MAC, a receiver-initiated MAC protocol, for communication among nanonodes in a wireless electromagnetic nanonetwork. The protocol can be used for a wide family of applications and operates in both distributed and centralized communication models. Furthermore, RIH-MAC is designed to operate adaptively with energy harvesting nanonodes. RIH-MAC is developed based on distributed and probabilistic schemes to create a scalable solution, which minimizes collisions and maximizes the utilization of harvested energy. Through simulation, we show the efficiency of RIH-MAC.

1. INTRODUCTION

Nanotechnology advancement promises to provide a significant rise in small scale communication. Wireless nanonetworks [2], [3] are a new generation of networks at nano scale, which are envisioned to be produced in the coming years. The nanonode of such a network is composed of nano antenna, nano memory, nano processing units, nano sensors, nano power storage, and so on. Each nanonode is in the range of nano to micro meters in size. Many new exciting applications are envisioned for nanonetworks. For example, nanosensors could detect chemical compounds at the molecular level or the presence of different infectious agents, such as viruses or harmful bacteria [3]. Many other applications can be imagined in various fields such as biology, medicine, chemistry, environmental, military, industrial, and consumer goods [3]. For example, nanosensors could be added to standard office products (pens, papers, etc.), making the idea of smart offices a reality.

The functionalities of nanonodes are realized only through communication. Nanosensors will collect useful information that must be sent outside of their sensing environment for storage and additional processing. Nanonodes need to communicate to control or actuate an action, or similarly monitor a phenomena. In other words, they will need to communicate between themselves as well as with nodes in the micro and macro domain. Among all possible communication methods for nanonodes (e.g. molecular communication, optical communication, acoustic communication), studies [2] show that electromagnetic communication in the 0.1-10.0 terahertz (THz) frequency band is a promising approach for communication in nanonetworks. We focus on the THz communication mechanism, as it can help nanosensors consume low energy while providing connectivity at the nano scale.

Due to the size limitation of nanonodes, only limited energy storage can be considered, where the nanonode harvests and stores energy from ambient resources. The tiny nanonodes will have a very limited energy storage capacity, probably in the form of ultra-nanocapacitors, which can only store enough energy for exchanging several hundred bits at a time [10]. Nanoscale harvesting elements such as nanowires [24] or biofuel cells [14], which enable harvesting from various resources such as vibration or blood sugar, will provide the possibility of communication for nanonodes. The variability of the energy resources makes the design of protocols difficult. For example, in the MAC layer, coordination between nanonodes is required to make sure that a nanonode receiver will have enough energy to receive packets from a nanonode transmitter at the moment of communication. The design of energy harvesting-aware solutions differs from traditional energy-aware protocols. Energy-aware protocols aim to minimize the consumption of energy while the energy harvesting-aware protocols aim to utilize the available energy. In nanonetworks, the energy is renewed, but the amount of available energy at each moment is limited. Thus, tailored energy harvesting-aware protocols for nanonetworks are required.

The problem of designing protocols to access the medium is not only difficult because of energy availability, but also because of special properties of nanonetworks. First, in most applications of nanonetworks, coordinating among hundreds of nanonodes is required. The tiny nanonodes are also limited in their processing capabilities. Therefore, complex protocols cannot be considered. Moreover, traditional MAC mechanisms such as message exchange or handshake for synchronization prior to data transfer should be minimized to reduce the consumption of energy as well as to enable the scalability of any solution. Due to these challenges, novel MAC protocols for nanonodes are required [12, 23].

This paper investigates the issue of MAC protocol design for nanonetworks, and develops a scalable, lightweight, distributed, and energy harvesting-aware solution. Unlike traditional MAC protocols, which mainly focus on minimizing collisions and bandwidth efficiency, our solution relies on a receiver-initiated communication model which addresses the matter of energy harvesting directly. In fact, a transmission occurs only if there is a high chance that the receiver will have enough energy for the reception. We develop two centralized and distributed types of our *receiver-initiated harvesting-aware MAC* (RIH-MAC). The centralized solution deals with topologies in which nanonodes are in direct communication with a more powerful device, called a nanocontroller [9, 23], which will be responsible for scheduling the communication with nanonodes. In the distributed RIH-MAC (DRIH-MAC), we develop a solution for an ad hoc formation of nanonodes. Each nanonode can communicate with the nanonodes in its neighborhood directly, and its neighbors provide connections to other nanonodes in the network. DRIH-MAC is more challenging since there is not a central point for scheduling communication. In both solutions, we include the properties of energy harvesting.

This is the first attempt to apply the idea of receiver-initiated transmission to energy harvesting nanonetworks. By coordinating the communication through the receiver in RIH-MAC, a transmitter adaptively selects its participation in network load, allowing RIH-MAC to achieve low collisions, a high packet delivery ratio, and high power efficiency. More specifically, our contributions take the following thrusts: (I) We present a probabilistic and distributed coordinated MAC protocol, RIH-MAC, employing receiver-initiated transmissions, in order to control medium access in a scalable and harvesting-aware fashion. (II) Due to the receiver-initiated design, RIH-MAC not only substantially reduces overhearing, but also achieves a lower collision probability. (III) RIH-MAC is applicable to a large family of nanonetwork applications and two network topologies: centralized and distributed.

Due to special characteristics of nanonetworks, traditional wireless MAC protocols (e.g. TDMA, CDMA, CSMA/CA) or sensor network protocols (e.g S-MAC [25], X-MAC [4]) are not applicable in the domain of nanonetworks. Recently, some MAC protocols have been proposed for electromagnetic nanonetworks [12, 23]. Jornet et al. proposed and analyzed a MAC protocol, PHLAME [12]. This protocol chooses the optimal value of code weight and repetition to address energy consumption and reliability. The performance of PHLAME is analytically studied in terms of energy consumption, delay, and achievable throughput. Later, Wang et al. [23] proposed an energy harvesting-aware and lightweight MAC protocol. However, the focus of the work is on the scheduling of packet transmissions by the nanocontroller, and thus it uses a centralized scenario. RIH-MAC, in contrast to previous MAC protocols for nanonetworks, is a receiver-initiated protocol that operates both in distributed and centralized topologies. Furthermore, RIH-MAC can adapt itself to various energy harvesting rates.

In the remainder of this paper, we first introduce the system model of nanonodes and characterize the nanonetwork in Section 2. Next, in Section 3, the RIH-MAC protocol is described, and in Section 4 it is evaluated through simulation. Related work is presented in Section 5, and finally the paper is concluded in Section 6.

2. SYSTEM MODEL

2.1 Physical Layer Communication Model

Nanonodes will communicate in the 0.1-10 THz frequency band [3, 10], which results in a micro to millimeter communication range [3, 9]. Communication in the THz band presents new channel properties: molecular absorption, thermal effect, etc. [9].

The nanonodes use pulse-based communication and Rate Division Time Spread On-Off Keying (RD TS-OOK) [11] as the modulation mechanism. A logical `1` is transmitted as a femto-second long pulse/symbol, and a logical `0` is transmitted as silence. Figure 1, for example, represents the modulation of `1011`.

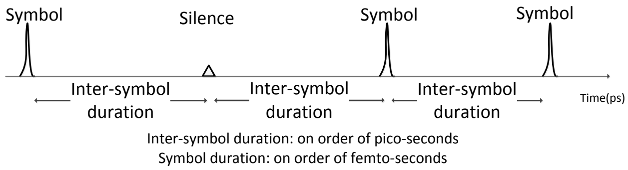


Figure 1: RD TS-OOK modulation for transfer of `1011`

The duration of each pulse is T_p and the time between two symbols is T_s, producing a symbol rate of $\beta = \frac{T_s}{T_p}$. The selection of optimal β is still an open question, though Wang et al. [23] investigated a solution for a centralized topology. It certainly depends on the hardware capabilities of the transmitter and receiver. Assuming there is no limitation in hardware capabilities, the existence of several flows of symbols from neighbor nanonodes may result in the collision of symbols. Moreover, energy availability is another factor that can affect the design of β. Currently, it is assumed that β takes values on the order of thousands.

2.2 Energy Model

Nanonodes need energy, mainly for their communication. Due to the limited size of nanonodes, they rely on harvesting methods, where nanoscale harvesters are required. Moreover, some of nanonode applications are designed for environments with no light or heat (e.g. inside the body, in liquid). Therefore, other sources of energy such as ambient vibration are considered [3] as the main method for energy harvesting. Advancements in nanowires and nanogenerators enable the production of nanoscale harvesters. A piezoelectric nanogenerator prototype [24] has shown promising results in harvesting energy from vibration at nanoscale. In piezoelectric harvesters, the amount of harvested energy depends on the vibration rate, not the acceleration amplitude. The variation in the vibration rate will result in a stochastic model for available energy for a nanonode at different times and different locations. Vibration in various environments represents a wide range of vibration rates [18, 19], e.g. from 1 Hz (person tapping his foot) to 2000 Hz (moving vehicle). In this work, we consider two scenarios: (I) when the energy harvesting rate is greater than the consumption rate; (II) when the energy harvesting rate is less than the consumption rate and follows a stochastic process. We show how RIH-MAC can adaptively operate in both scenarios. Moreover, we consider an ultra nano-capacitor with non-linear behavior as the energy storage of each nanonode [10].

2.3 Network Model

We consider two models for a network of nanonodes: centralized and distributed. In the centralized model, a central node, called a nanocontroller [23], is responsible for coordination among nanonodes. All traffic generated by nanonodes will be transmitted to the nanocontroller, and then the nanocontroller is responsible for transfering it to the micro and macro domains. The second model, namely distributed, is an ad hoc network of nanonodes, where each nanonode can only communicate with its neighbors, i.e., nanonodes in communication range. The nanonodes are responsible for forwarding the traffic of their neighbors. The forwarding mechanism is out of the scope of this work. In both models, we are assuming that the topology would be static, i.e., nanonodes have no mobility model.

2.4 Application Requirements

We assume the applications for nanonodes are delay tolerant. This assumption particularly applies to scenarios where the energy harvesting rate is lower than the consumption rate. In the THz band, the available bandwidth is very large (e.g. hundreds of gigabits per second). Therefore, the delay in packet transmission and propagation is on the order of picoseconds. The only delay imposed is from the time required to harvest enough energy to exchange packets.

Furthermore, applications are not loss sensitive. Therefore, we consider only a simple acknowledgement scheme and a limited number of retries for unsuccessful transmissions. This will be the main mechanism to compensate for packet loss due to molecular absorption and thermal noise. It also handles any loss due to collisions of packets. We mainly reduce the probability of collisions as part of our MAC design as will be discussed later in Section 3.

Moreover, we are assuming that the packets are generated at a constant rate. Also, in the distributed network model, we assume that a forwarding mechanism is designed in a way that the forwarding traffic rate would be almost equal for all nanonodes. Therefore, the packet transmission and reception rates of all nanonodes are almost equal.

Finally, in scenarios with limited available energy (the harvesting rate is lower than the consumption rate), the packet generation rate is designed in a way that there would not be any packet overflow at the source or intermediate nodes.

3. RECEIVER-INITIATED COMMUNICATION

Our communication model between nanonodes is receiver-initiated. Time is divided into equal timeslots. In each timeslot, two packets are exchanged between a sender and one of several receivers. The receiver announces to one or several nanonodes with a *ready to receive* (RTR) packet, which means it is ready to receive a packet. The recipient of the *RTR* packet may transmit a *DATA* packet accordingly. If required, the receiver can transmit a corresponding *ACK* in the next *RTR* packet.

Figure 2 illustrates a sample sequence of *RTR* and *DATA* packets between a receiver and a sender. When the first *RTR* is transmitted, the sender does not receive it, which could be for many reasons, e.g., lack of energy, communicating with another node. In the next slot, the sender receives the *RTR* packet, but does not transmit a *DATA* packet, which again could be due to many reasons, e.g., lack of energy. Upon receiving the third *RTR*, the sender transmits a *DATA* packet and the receiver receives it. The detail of scheduling of when to transmit and receive *RTR*s is part of RIH-MAC, which will be described later in this section. The *RTR* packet contains the node id, destination id (0 for broadcast), number of neighbors, maximum known degree, current amount of energy, mode of communication (centralized or distributed), and other fields that will be described in the reminder of this section.

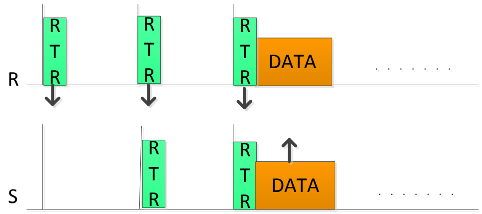


Figure 2: *RTR* and *DATA* Packets between a (R)ecevier and a (S)ender

There are two reasons for choosing the receiver-initiated communication model. First, in a centralized topology, the nanocontroller is responsible for the management of communication among nanonodes. Due to the higher energy budget of the nanocontroller and the need for more efficient usage of energy on the transmitter side, the receiver-initiated communication model moves the load of energy consumption for the management of communication and packet handling to the nanocontroller. Furthermore, since it is assumed there are abundant nanonodes, a significant portion of them may not be able to transmit a packet at each time slot. So, the receiver-initiated method enables the chance of having a fair traffic flow from different nanonodes while it does not need to be concerned about the energy level of nanonodes, as will be described in Section 3.1.

Second, independent of the communication model (centralized or distributed), it is better to initiate communication only when it is most probable that the receiver will have enough energy to receive a packet. Otherwise, many transmissions would be unsuccessful because of a high chance of the receiver not having enough energy. Note that handshaking does not look to be an efficient method for the small packet sizes that nanonodes can handle. However, there is still a need for scheduling, which is more complex for the distributed communication model. We will introduce our scheduling model for the distributed model in Section 3.2.

3.1 Centralized

In the centralized model, a nanocontroller receives information from nanonodes and then forwards it for further processing in the micro and macro domains. This model is valid in many applications where nanosensors collect information about their target phenomena. This model has been also used by other work [17, 23] and is the simplest and most scalable method to develop a nanonetwork.

To collect information, the nanocontroller repetitively broadcasts *RTR* packets. After each RTR, one or several nanonodes may transmit a *DATA* packet. The decision about which nanonode transmits its *DATA* packet follows a random process, where an arbitrary nanonode will participate with probability p. The nanonode will have enough energy for the reception of the *RTR* packet and the transmission of the consequent *DATA* packet with probability q. Also, the nanonode may have a *DATA* packet to transmit with probability r. Then, upon the reception of *RTR* packet, we want the nanonodes to participate in transmitting a *DATA* packet in a way that only one of them transmits. This will avoid collisions due to simultaneous transmissions which result in the waste of timeslots and energy. The expected number of concurrent transmissions X by the n nanonodes can be written as

$$E[X] = p \cdot q \cdot r \cdot n\,. \tag{1}$$

Setting the expected value equal to 1 will indicate the probability of participation to transmit a *DATA* packet by each node as

$$p = \frac{1}{q \cdot r \cdot n}\,. \tag{2}$$

The next *RTR* can contain the corresponding *ACK* for the transmitted *DATA* packet. This way, the participating nanonode can infer any possible collision or packet loss for retries. Furthermore, note that with the assumption of fixed size *RTR* and *DATA* packets, each nanonode knows the beginning of each timeslot for later transmissions, just after the reception of the first *RTR*.

The centralized model is scalable for a large number of nanonodes. Also, in the case of no energy constraint or a few number of nanonodes, RIH-MAC can provide a high data rate. For example, transferring a terabyte piece of information between two devices could be achieved by placing them in close proximity to each other.

In some scenarios, it is required to transmit data from the nanocontroller to nanonodes, e.g. updating the functionality of nanonodes. For down-link, i.e., transmitting data from the nanocontroller to nanonodes, the same mechanism as uplink is used with a minor change in one field of the *RTR* packet. In this scenario, the *dir* field of the *RTR* packet is set to 1, which means that the nanocontroller is not expecting a *DATA* packet from nanonodes and instead will transmit a *DATA* packet. The nanonode that receives this *RTR* waits to receive the consequent *DATA* packet. The only overhead of this method is that in an energy limited scenario, this *DATA* should be sent several times until all nanonodes receive it. Assuming a similar model of participation as uplink, a *DATA* packet should be transmitted at least n times to make sure that the expected number of nanonodes that receive the *DATA* packet is n.

3.2 Distributed

A distributed ad hoc formation of nanonodes looks to be unavoidable in many situations, e.g. when the nanocontroller cannot be in direct communication with all nanonodes. Here, we extend our RIH-MAC to support the ad hoc formation of nanonodes. As before, the communications are receiver-initiated, and the nanonodes may not necessarily have enough energy for communication at all timeslots.

Common random access methods such as CSMA/CA and their handshake extensions, e.g. RTS/CTS, are not applicable in nanonetworks for several reasons. First, handshaking is a heavy-weight process for nanonodes. Second, synchronization and lack of energy makes the handshake process inefficient for nanonodes. Therefore, new access mechanisms are required [12, 23].

Our medium access method relies on the receiver-initiated principle and distributed scheduling for nanonodes, which is energy-efficient, energy-adaptable, lightweight, and scalable. Energy adaptable means that scheme is adaptable to the various energy harvesting rates. Our scheme uses distributed scheduling for communication among nanonodes. Communication between a group of ad hoc formed nanonodes can be modeled as an edge-coloring problem. Each pair of nanonodes that are in communication range of each other will have an edge between them. All incident edges of a node should have different colors. Each color represents the timeslot in which node can communicate with one of its neighbors.

The edge coloring problem is NP-complete, and by Vizing's theorem, the number of colors needed to edge color a graph is either its maximum degree Δ or $\Delta + 1$. Most edge coloring solutions are centralized. Here, we are looking for a lightweight distributed solution. Among distributed solutions, we adopt a solution by Grable and Panconesi [7] with minor changes. This method can color a graph with $(1+\epsilon)\Delta$ colors, for any positive ϵ in $O(\log\log n)$ rounds, where n is the number of vertexes. The method finds a coloring solution for the problem with a high probability close to 1. Most other distributed and deterministic models such as [5] have too much overhead to run on nanonodes, and also their performance improvement is not significant. However, this algorithm satisfies the simplicity and distributed properties that we require. When this scheme fails to color properly, it can be run again at a low cost. Note that even though a network of nanonodes will be mainly static, its formation and topology can be dynamic over time (due to failure of nanonodes, or adding or removing some nanonodes), and therefore coloring will need to be run again.

Our distributed edge coloring algorithm is shown in Algorithm 1. Each edge $w = (u, v)$ between two arbitrary nanonodes u and v is initially given a palette of $(1+\epsilon) \cdot \max(deg(u), deg(v))$ colors, where $deg()$ denotes the degree of the node. This palette is recorded locally at each nanonode. The formation of this palette is also done through receiving and transmitting some initial *RTR* packets where no *DATA* packets are sent in reply. A new nanonode that has no color assigned for its edges will transmit zero in the *color* field of its *RTR* packet. The main coloring process occurs in rounds. In each round, each uncolored edge independently picks a tentative color uniformly at random from its current palette. If no other edges of nodes u and v are using this color, it is picked as the final color of edge w. Otherwise, the coloring of this edge will be tried again in the next round. At the end of each round, the palettes are updated in the obvious way: colors successfully assigned are deleted from the current palette. The duration of each round would be equal to the exchange of *RTR* packets to announce the selected colors and receiving the selected colors from neighbors. Therefore, to reach the agreement or disagreement on a color with all Δ neighbors through *RTR* packets, at most $\Delta + 1$ timeslots is required for each round with the assumption of no *RTR* packet failure. More rounds are required for these circumstances. A colored graph is illustrated in Figure 3. Colors are also labeled with numbers.

Each link between two nanonodes is bidirectional. One way to schedule the direction of communication is to extend Algorithm 1 to assign two colors per edge. However, since we assume a nanonode cannot transmit and receive at the same time, it would be similar to switching between the transmission and reception states, consecutively. For simplicity, we assume consequent changes of the communication direction

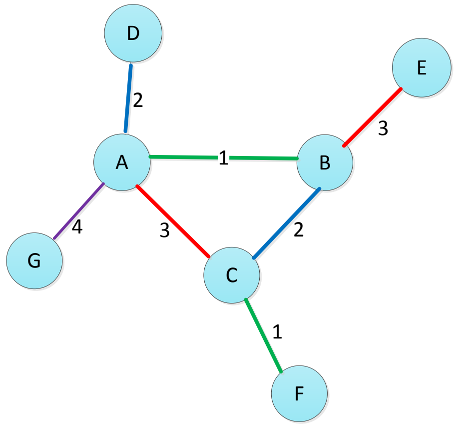


Figure 3: A Colored Graph. Here each number represents a different color.

as shown in Figure 4. A node with a lower id, here alphabetically ascending, sends in the first slot and receives in the following slot for each link. For example, for the link with color 2 between nodes B and C, first B plays the role of sender at slot 3 (depicted as 2S) and C plays the role of receiver (depicted as 2R). In the next slot (4), B receives (depicted as 2R) and C transmits (depicted as 2S). Recall that the exchange of a *RTR* and *DATA* packet occurs in each timeslot with the receiver initiating it. Note that slots 7 and 8 are not used by B and C. It may look to be a waste of slots. However, this is the cost to pay for communication without collisions.

Algorithm 1: Coloring Algorithm for DRIH-MAC

```
Void Color()
    output: Colors for each link
    Estimate the number of neighbors by listening to RTR packets;
    Announce my presence to neighbors with RTR packets;
    For link $w$ between $u$ and $v$, select a palette of colors with $d = \max\{d(u), d(v)\}$ colors;
    while $w$ with unknown color
        select one color randomly from palette;
        if color is the same for $w$ by both $u$ and $v$
            Finalize the color;
```

Distributed RIH-MAC avoids collisions due to concurrent transmissions by exploiting the coloring mechanism. DRIH-MAC is preferred to the random access methods. First, the traffic rate of nanonodes are very similar to each other, so, there is no need to provide more access to the medium for one nanonode over another. Second, DRIH-MAC is adaptable to energy limited scenarios. With no coordination before the transmission, a transmitter can understand if the receiver has enough energy to receive the packet. Furthermore, in scenarios where the energy harvesting rate is lower than the consumption rate, some slots eventually will not be utilized due to lack of energy. An optimum energy consumption mechanism can coordinate its communication schedule with these empty slots to maximize energy and timeslot utilization.

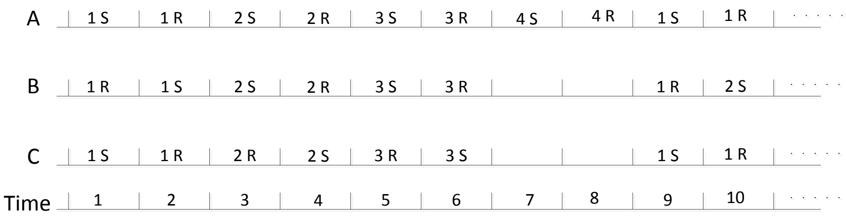
DRIH-MAC still suffers from the *hidden terminal problem*. For example, when A is transmitting to D and B is transmitting to C, there could be problem at C in distinguishing pulses from B and A. Low code weights can be one approach to mitigate this problem. Another approach is to select the direction of communication to avoid this problem. Nevertheless, finding the best approach is part of our future work.

3.3 Energy Consumption Schedule

Distributed RIH-MAC can be run stand-alone if there is no energy limitation on nanonodes. However, a coordinated energy consumption schedule (CECS) between two communicating nanonodes is required to achieve the highest performance of DRIH-MAC. When there is no such coordination, many *RTR* packets would be sent with no *DATA* packet response. Similarly, transmitters may listen for *RTR* packets but receive no *RTR* packets. In both scenarios, energy is wasted. Here, we describe our prediction-based CECS. Since the process of energy harvesting for neighbor nanonodes is not known exactly, the prediction acts based on the amount of available energy of neighbors during the previous slot (which has been received in *RTR* packets) and a predefined consumption model. While CECS is a probabilistic approach, it improves energy consumption significantly.

We assume that nanonodes follow a similar harvesting model. The amount of current energy is received from each neighbor through *RTR* packets. Recall also that *RTR* packets contain the number of neighbors. We are assuming there is an *optimum policy*, which specifies for each nanonode how much energy should be spent per level of energy. Such a model can be solved by modeling the problem as a Markov decision process, and then the offline result can be stored as a lookup table [6].

Furthermore, all nanonodes follow the same consumption model. The amount of consumption per *cycle* of consumption is determined based on the designed *optimum policy*. Each *cycle* is a round of passing through all timeslots for all neighbors, in which the nanonode may transmit or receive messages. For example, in one cycle for a nanonode with 5 neighbors, with the current level of energy, it is determined that only two packets can be received from neighbors. If those two nanonodes are neighbors 2 and 4, then the consumption pattern for that cycle from left to right would be `01010`. To provide a fair traffic flow among all neighbors, this pattern rotates at the end of each cycle. For example, the mentioned pattern after one cycle would be `00101`. All nanonodes will use the same pattern for different levels of energy. For example, Table 1a shows the pattern for five links in the first cycle. The pattern can be extended for various numbers of links to keep a balance for all links, e.g., Table 1b. The nanonode will alternate between the pattern choices for the same policy, e.g., two patterns for policy 2 in Table 1b. Although nanonodes follow the same pattern, they will be independent in their own rotation. The *rotation offset number* for each nanonode is transferred in the *RTR* packets. Moreover, the patterns for the transmission and the reception are independent. A receiver decides to transmit its *RTR* if it predicts that the transmitter has a schedule to receive the *RTR* based on the previous received rotation off-



A	1 S	1 R	2 S	2 R	3 S	3 R	4 S	4 R	1 S	1 R	· · · · ·
B	1 R	1 S	2 S	2 R	3 S	3 R			1 R	2 S	· · · · ·
C	1 S	1 R	2 R	2 S	3 R	3 S			1 S	1 R	· · · · ·
Time	1	2	3	4	5	6	7	8	9	10	· · · · ·

Figure 4: Example Communication in DRIH-MAC. The nanonodes A, B and C from Figure 3 are shown. S indicates the sending mode, and R indicates receiving mode. The number preceding S/R indicates the *color*.

set number. However, since this prediction can be incorrect, some *RTRs* can still be wasted, and consequently no *DATA* reply is received. This is avoidable only if the nanonodes decide about their energy consumption optimization model together, which looks to be implementable with significant overhead. Therefore, here we do not evaluate such a solution. At each timeslot of a cycle, the transmitter S waits to receive a *RTR* from the receiver only if based on the schedule, it is expecting a *RTR* from the receiver. Similarly, a receiver will transmit a *RTR* only if based on the transmitter schedule, it predicts that the transmitter will be waiting for a *RTR* to send its *DATA*. Note that these controls and predictions are simple enough to run on a nanonode. Through this method, the transmitter does not consume energy for the reception of *RTR* when one is not sent. Also, the receiver will not transmit any *RTR* if it predicts that the transmitter is not scheduled to receive the *RTR* and send a *DATA* packet.

Table 1: Patterns Corresponding to Various Policies (policy number is equal to the number of receptions in one cycle)

(a) Pattern for 5 Links

Pattern	Policy
0	0 0 0 0 0
1	1 0 0 0 0
2	0 1 0 1 0
3	1 0 1 0 1
4	0 1 1 1 1
≥ 5	1 1 1 1 1

(b) Pattern for 4 Links

Pattern	Policy
0	0 0 0 0
1	1 0 0 0
1	0 1 0 0
2	0 1 0 1
2	1 0 1 0
3	1 0 1 1
3	0 1 1 1
≥ 4	1 1 1 1

A detailed analysis of ensuring that there will exist slots in which both transmitter and receiver will be scheduled to send and receive at the same time is too lengthy to be included here. Briefly, it can be described as follows. When the transmitter and receiver do not happen to be in `1` at the same time, they will jump into other states of energy due to changes in the consumption and harvesting of energy. In the worst scenario, both transmitter and receiver will eventually go to the last policy and its corresponding pattern, i.e., all `1`, which will certainly result in communication. To make it more clear, we show the measurements in simulation results, which numerically analyze the performance of CECS.

4. PERFORMANCE EVALUATION

We ran several experiments to evaluate the performance of RIH-MAC. For our simulation, we modified and enhanced the *Nanosim* module [17], which enables simulation of electromagnetic nanonetworks in ns-3. The major modifications were the energy module and channel model. Nanonodes have harvesters that follow the harvesting model developed in [10]. To evaluate the effect of harvest rate, we characterize the harvest rate as a probability distribution function, where it is discretized to adapt to the simulation environment. Each nanonode has an ultra-nanocapacitor as the energy storage with 100 picojoule capacity.

Nanonodes are considered to be operating in an environment with 10% water vapor with the corresponding channel path loss model [10] in the 100-300 GHz frequency band. Energy consumption is modeled as 1 femtojoule for the transmission of each pulse and 0.1 femtojoule for the reception of each pulse [10, 16, 23]. The size of packets is selected based on the method we developed in [15], where we model and find the optimum packet size for several optimization functions. There is always a back-log of packets ready in a queue to transmit. We present the results of simulation for the centralized and distributed RIH-MAC in the following sections.

4.1 Centralized

In this scenario, nanonodes are distributed in a sphere with a radius of 10 mm. A nanocontroller is placed in the center. The nanonodes can communicate directly with the nanoncontroller. Every 100 ms, the nanocontroller transmits a *RTR* packet and waits for the reception of a *DATA* packet from one of the nanonodes. Nanonodes decide on their probability of transmitting a *DATA* packet based on (2). Figure 5 illustrates the percentage of time the nanocontroller receives a *DATA* packet. The theory and simulation results are very close. As can be seen, RIH-MAC is scalable, i.e., with the growth in nanonodes, the percentage of *DATA* receptions remains almost the same. Also, as illustrated in Figure 6, the probability of collision (i.e., simultaneous transmission of two or more nanonodes) becomes almost constant with an increase in the number of nanonodes.

4.2 Distributed

In this scenario, nanonodes are distributed uniformly in a cube of size 100 × 100 × 10 mm. Before evaluating the performance of the CECS, we first show the performance of edge coloring. We want to show (I) the probability of successful coloring and (II) the time it takes to color. Figure 7 shows the probability of successful coloring of the nanonode graph for various values of ϵ. As can be seen for all values, the probability of success is more than 99%, and the higher ϵ, the higher the probability of successful coloring.

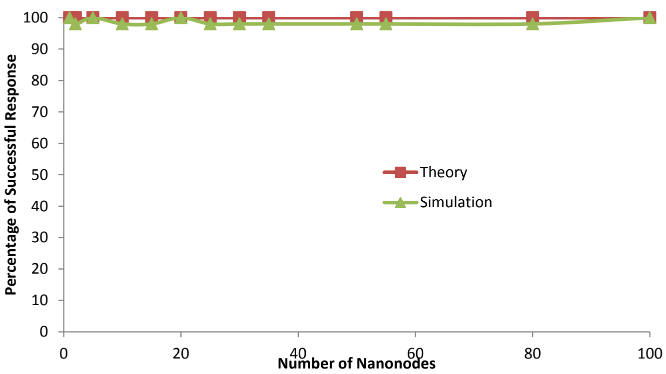


Figure 5: Percentage of Receiving a *DATA* Packet in Response to a *RTR* Packet

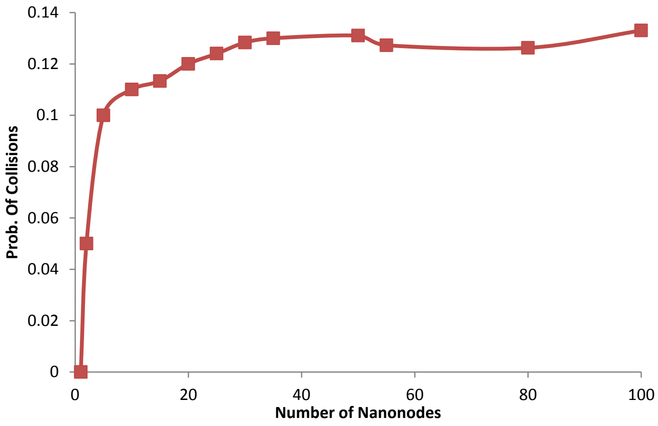


Figure 6: Probability of Collisions

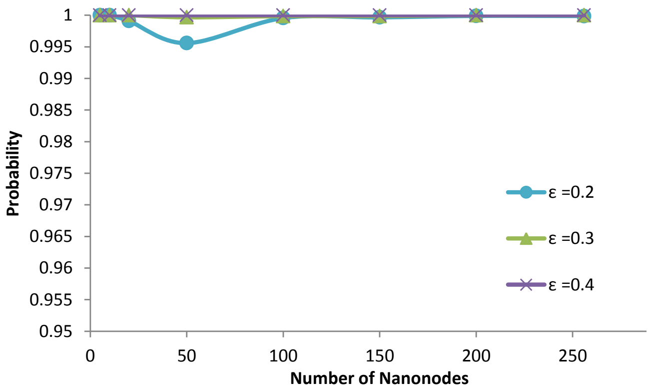


Figure 7: Probability of Successful Edge Coloring

Figure 8 depicts the number of rounds required until all edges are colored properly. Clearly, for a higher number of nanonodes, it takes more rounds to color, but it still is a reasonable number of rounds. Recall that the duration of one round is equal to the exchange of $2 \cdot (\Delta + 1)$ *RTR* packets. Since the duration of *RTR* packets is very short, the scheme converges quickly, e.g. less than one nanosecond in the scenario with no energy limit and 256 nanonodes.

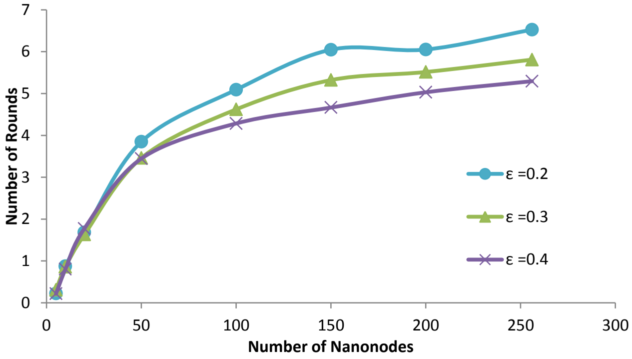


Figure 8: Number of Rounds to Color Edges

To evaluate the performance of CECS, we define the following metric.

$$RTR\_Success = \frac{RTR_c}{RTR_c + RTR_u + RTR_w},$$

where RTR_c is the number of *RTRs* with a successful *DATA* response, RTR_u is the number of *RTRs* which are not heard by the targeted sender due to lack of energy, and RTR_w is the number of *RTRs* which are received, but cannot be replied to due to lack of energy. Note that the value of RTR_w for CECS is zero since a nanonode will not listen to *RTRs* if it knows that it will not have energy for transmission.

Figure 9 illustrates the performance of CECS in comparison with the scenario where there is no scheduling for the transmission of *RTR*s. CECS achieves close to 100% success as the harvesting rate increases. The no-CECS case has a slower slope of improvement.

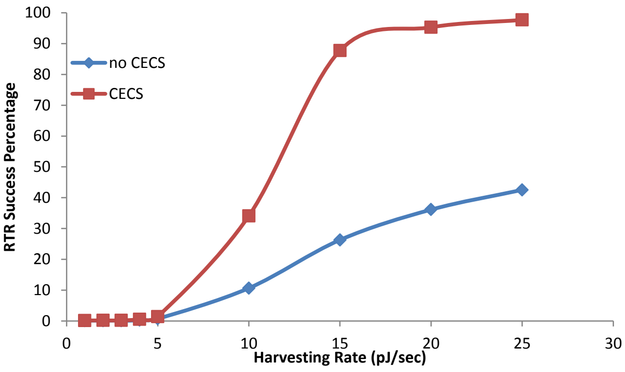


Figure 9: *RTR* Success Percentage with Poisson Energy Arrival

In general, as the harvesting rate is increased, *RTR_Success* becomes closer to 100% because energy would exist at all times, and RTR_u becomes zero. This observation can also be seen in Figure 10, where the no-CECS scheme becomes closer to the CECS faster for the lognormal distribution of energy arrival as compared to the Poisson distribution used in Figure 9.

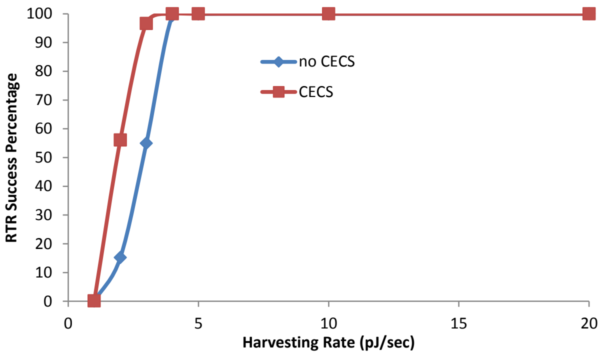


Figure 10: RTR Success Peercentage with Lognormal Energy Arrival - $\sigma^2 = 0.5 \cdot \mu$

Finally, we measure the fairness index for the communication with neighbors. As can be viewed in Figure 11, CECS achieves a better fairness index than the random selection of neighbors at each cycle.

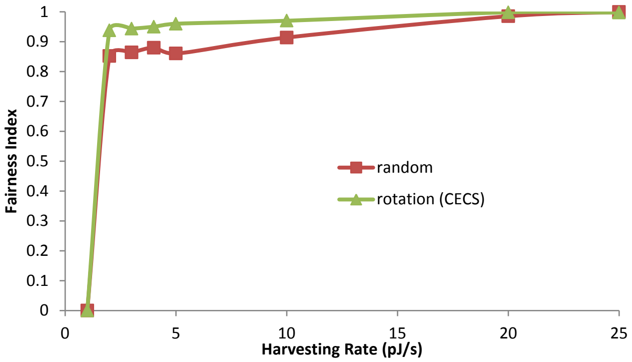


Figure 11: Fairness Index vs. Harvesting Rate - Poisson Energy Arrival

Furthermore, it can be observed that with an increase in the harvesting rate, fairness is increased, which actually occurs because of a more successful chance of message reception. The fairness index, indeed, confirms that not only will CECS result in communication between a nanonode with all of its neighbors, but it will do so in a balanced fashion.

5. RELATED WORK

The protocol design for nanonetworks is in its early stages. In our previous work [15], we introduced an optimization model to find the optimal values for packet size, code weight, and repetition. We designed a multiobjective function problem to address several functions such as energy consumption, end to end delay, and communication reliability. Jornet and Akyilidiz proposed PHLAME [12], which mainly focuses on applying the optimal values of code weight and repetition for packet transmission. The performance of the proposed protocol is analytically studied in terms of energy consumption, delay, and achievable throughput, by using models of the Terahertz channel. However, implementation feasibility and energy efficiency evaluation of the method are still open questions. Later, Wang et al. [23] proposed an energy harvesting-aware and light-weight MAC protocol. The protocol attempts to achieve fair throughput and optimal channel access among nanosensors which are controlled by a nanocontroller. Towards this end, the critical packet transmission ratio is defined, which is the maximum allowable ratio between the transmission time and the energy harvesting time. A nanosensor has to harvest more energy than it consumes to reach perpetual data transmission. However, this protocol does not design and evaluate the distributed model of communication among nanonodes.

There has been a large body of research in MAC protocol design for sensor networks and UWB networks. A comprehensive survey was compiled by Akyildiz et al. [1] for sensor networks and by Gupta and Mohapatra [8] for UWB networks. However, these MAC protocols cannot directly be used in nanonetworks because they do not consider either the limitations of nanodevices, energy harvesting, or the characteristics of the Terahertz band. Moreover, the majority of existing MAC protocols for wireless networks have been designed for band-limited channels. In nanonetworks, the Terahertz channel provides nanodevices with an almost 10 THz wide window.

Furthermore, carrier-sensing techniques in classical MAC protocols cannot be used in pulse-based communication systems since there is no carrier for sensing. Only some solutions [8] proposed for Impulse Radio Ultra Wide Band (IR-UWB) networks could be considered, but their complexity limits their usefulness in the nanonetwork scenario. For example, generating and distributing orthogonal time hopping sequences is not a lightweight process for nanodevices. Moreover, the characteristics of the THz band as well as the limited processing capabilities of nanodevices are the major factors that necessitate the redesign of protocols for the networking of nanonodes.

The main limitation for nanodevices results from the limited energy that can be stored in nanobatteries or nanocapacitors. Therefore, energy harvesting-aware protocols are required. Recently, energy harvesting-aware design for sensor networks has been studied. However, most of the studies cannot be applied to nanonetworks. First, the energy storage of nanonodes is limited while in previous work, it is mainly considered infinite or extremely large. Second, most of the schemes (e.g.,[13, 20]) are too complex to run on nanonodes. Finally, the energy harvesting rate is usually considered very close the consumption rate in previous work. However, in nanonetworks, the harvesting rate, for most energy resources, is smaller than the energy consumption rate. This needs to be considered in the design of nanonetworks.

Receiver-initiated protocols have been investigated in duty cycle sensor networks [21, 22]. However, those methods cannot be used directly for energy harvesting environments due to the stochastic properties of energy harvesting. Moreover, it is not clear how much these receiver-initiated protocols can be effective in energy harvesting-aware protocols. In this work, we investigated the use of receiver-initiated protocols for energy harvesting nanonetworks.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a receiver-initiated MAC protocol for electromagnetic wireless nanonetworks. Nanonodes of such a network rely on energy harvesting to supply energy for their communication. Our receiver-initiated protocol, RIH-MAC, takes into account the energy harvesting properties of nanonodes, where they may form a centralized or distributed network. RIH-MAC is scalable with the increase in the number of nanonodes and also leads to a low number of collisions. This protocol is adaptable to be deployed in a large family of nanonetwork applications, where delay and packet loss are not hard QoS requirements.

In future work, we are planning to first solve the hidden terminal problem, where concurrent transmissions among neighbor nanonodes can affect each other's successful packet receptions. Furthermore, we will further evaluate the effect of various energy harvesting rates on network performance metrics such as delay and bandwidth.

7. REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Comm. Magazine*, 40(8):102 – 114, Aug. 2002.

[2] I. F. Akyildiz, F. Brunetti, and C. Blazquez. Nanonetworks: A new communication paradigm. *Computer Networks*, 52(12):2260–2279, 2008.

[3] I. F. Akyildiz and J. M. Jornet. Electromagnetic wireless nanosensor networks. *Nano Comm. Networks*, 1(1):3–19, 2010.

[4] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proc. of the 4th ACM SenSys*, SenSys, pages 307–320, 2006.

[5] S. Gandham, M. Dawande, and R. Prakash. Link scheduling in wireless sensor networks: Distributed edge-coloring revisited. *Journal of Parallel and Distributed Computing*, 68(8):1122 – 1134, 2008.

[6] M. Gorlatova, A. Wallwater, and G. Zussman. Networking low-power energy harvesting devices: Measurements and algorithms. *IEEE Trans. on Mobile Computing*, 12(9):1853–1865, 2012.

[7] D. A. Grable and A. Panconesi. Nearly optimal distributed edge colouring in O(log log n) rounds. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 278–285, Philadelphia, PA, USA, 1997.

[8] A. Gupta and P. Mohapatra. A survey on ultra wide band medium access control schemes. *Comput. Netw.*, 51(11):2976–2993, Aug. 2007.

[9] J. Jornet and I. Akyildiz. Channel capacity of electromagnetic nanonetworks in the terahertz band. In *IEEE International Conference on Communications (ICC)*, pages 1 –6, May 2010.

[10] J. Jornet and I. Akyildiz. Joint energy harvesting and communication analysis for perpetual wireless nanosensor networks in the terahertz band. *IEEE Trans. on Nanotechnology*, 11(3):570–580, May 2012.

[11] J. M. Jornet and I. F. Akyildiz. Information capacity of pulse-based wireless nanosensor networks. In *Proceeding of IEEE SECON*, pages 80–88, 2011.

[12] J. M. Jornet, J. C. Pujol, and J. S. Pareta. PHLAME: A physical layer aware MAC protocol for electromagnetic nanonetworks in the terahertz band. *Nano Communication Networks*, 3(1):74 – 81, 2012.

[13] R.-S. Liu, K.-W. Fan, Z. Zheng, and P. Sinha. Perpetual and fair data collection for environmental energy harvesting sensor networks. *IEEE/ACM Trans. on Networking*, 19(4):947 –960, Aug. 2011.

[14] K. MacVittie, J. Halamek, L. Halamkova, M. Southcott, W. D. Jemison, R. Lobel, and E. Katz. From "cyborg" lobsters to a pacemaker powered by implantable biofuel cells. *Energy Environ. Sci.*, 6:81–86, 2013.

[15] S. Mohrehkesh and M. C. Weigle. Optimizing communication energy consumption in perpetual wireless nanosensor networks. In *Proceedings of IEEE Globecom*, Atlanta, GA, December 2013.

[16] T. Otsuji, S. Tombet, A. Satou, M. Ryzhii, and V. Ryzhii. Terahertz-wave generation using graphene: Toward new types of terahertz lasers. *IEEE Journal of Selected Topics in Quantum Electronics*, 19(1), 2013.

[17] G. Piro, L. A. Grieco, G. Boggia, and P. Camarda. Nano-sim: simulating electromagnetic-based nanonetworks in the network simulator 3. In *Proc. of Workshop on NS- 3*, Cannes, France, Mar. 2013.

[18] S. Roundy. On the effectiveness of vibration-based energy harvesting. *Journal of Intelligent Material Systems and Structures*, 16(10):809–823, 2005.

[19] S. Roundy, P. K. Wright, and J. Rabaey. A study of low level vibrations as a power source for wireless sensor nodes. *Computer Communications*, 26(11):1131 – 1144, 2003.

[20] V. Sharma, U. Mukherji, V. Joseph, and S. Gupta. Optimal energy management policies for energy harvesting sensor nodes. *IEEE Transactions on Wireless Communications*, 9(4):1326 –1336, apr 2010.

[21] Y. Sun, O. Gurewitz, and D. B. Johnson. RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of the 6th ACM SenSys*, pages 1–14, 2008.

[22] L. Tang, Y. Sun, O. Gurewitz, and D. Johnson. PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks. In *Proceedings of IEEE INFOCOM*, pages 1305–1313, 2011.

[23] P. Wang, J. M. Jornet, M. A. Malik, N. Akkari, and I. F. Akyildiz. Energy and spectrum-aware MAC protocol for perpetual wireless nanosensor networks in the terahertz band. *Ad Hoc Networks*, 11(8):2541 – 2555, 2013.

[24] S. Xu, B. J. Hansen, and Z. L. Wang. Piezoelectric-nanowire-enabled power source for driving wireless microelectronics. *Nat Commun*, 1, Oct. 2010.

[25] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proc. of IEEE INFOCOM*, pages 1567–1576, 2002.