# Joint physical and link layer error control analysis for nanonetworks in the Terahertz band

N. Akkari[1] · J. M. Jornet[3] · P. Wang[4] · E. Fadel[1] · L. Elrefaei[1] · M. G. A. Malik[1,2] · S. Almasri[1,2] · I. F. Akyildiz[1,5]

Published online: 4 August 2015

**Abstract** Nanonetworks consist of nano-sized communicating devices which are able to perform simple tasks at the nanoscale. The limited capabilities of individual nanomachines and the Terahertz (THz) band channel behavior lead to error-prone wireless links. In this paper, a cross-layer analysis of error-control strategies for nanonetworks in the THz band is presented. A mathematical framework is developed and used to analyze the tradeoffs between Bit Error Rate, Packet Error Rate, energy consumption and latency, for five different error-control strategies, namely, Automatic Repeat reQuest (ARQ), Forward Error Correction (FEC), two types of Error Prevention Codes (EPC) and a hybrid EPC. The cross-layer effects between the physical and the link layers as well as the impact of the nanomachine capabilities in both layers are taken into account. At the physical layer, nanomachines are considered to communicate by following a time-spread on-off keying modulation based on the transmission of femtosecond-long pulses. At the link layer, nanomachines are considered to access the channel in an uncoordinated fashion, by leveraging the possibility to interleave pulse-based transmissions from different nodes. Throughout the analysis, accurate path loss, noise and multi-user interference models, validated by means of electromagnetic simulation, are utilized. In addition, the energy consumption and latency introduced by a hardware implementation of each error control technique, as well as, the additional constraints imposed by the use of energy-harvesting mechanisms to power the nanomachines, are taken into account. The results show that, despite their

This project was funded by the National Plan for Science, Technology and Innovation (MAARIFAH)-King Abdulaziz City for Science and Technology- the Kingdom of Saudi Arabia- award number 12-NAN230-03. The authors also acknowledge with thanks the Science and Technology Unit, King Abdulaziz University for technical support.

✉ J. M. Jornet
jmjornet@buffalo.edu

N. Akkari
nakkari@kau.edu.sa

P. Wang
pu.wang@wichita.edu

E. Fadel
efadel@kau.edu.sa

L. Elrefaei
laelrefaei@kau.edu.sa

M. G. A. Malik
mgmalik@kau.edu.sa; mgmalik@uj.edu.sa

S. Almasri
smalmasri@kau.edu.sa; smalmasri@uj.edu.sa

I. F. Akyildiz
ian@ece.gatech.edu

[1] Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

[2] Faculty of Computing and Information Technology, University of Jeddah, Jeddah, Saudi Arabia

[3] Department of Electrical Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260, USA

[4] Department of Electrical Engineering and Computer Science, Wichita State University, Wichita, KS 67260-0083, USA

[5] Broadband Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

simplicity, EPCs outperform traditional ARQ and FEC schemes, in terms of error correcting capabilities, which results in further energy savings and reduced latency.

**Keywords** Nanonetworks · Terahertz band · Error control · Pulse-based communication

## 1 Introduction

Nanotechnology is providing the engineering community with a new set of tools to design and manufacture novel nanoscale devices, which are able to perform simple tasks, such as computation, data storing, sensing and actuation. The integration of several of these nano-devices into a single entity will enable the development of more advanced nanomachines. By means of communication, these nanomachines will be able to achieve complex tasks in a distributed manner [2]. The resulting nanonetworks will enable unique applications in the biomedical, industrial and military fields, such as in-vivo health monitoring systems, nanosensor networks for biological and chemical attack prevention, or massive multi-core processing architectures on chip [1].

Recent developments in the area of graphene-based nano-electronics point to the Terahertz (THz) band (0.1–10 THz) as the frequency band of communication for nano-devices [4, 12]. The THz band provides nanomachines with an unprecedentedly large bandwidth, which ranges from several tens of GHz up to a few THz, and enables data transmissions at multi-Gigabits-per-second (Gbps) or even Terabits-per-second (Tbps) [10, 17]. However, this comes at the cost of a very high propagation loss, which given the power constraints of energy-harvesting nanomachines [3, 11], results in a very short communication distance, usually much below 1 m.

The low-power communication constraints of nanomachines aggravate the effects of the THz-band channel and lead to error-prone wireless links. The performance of traditional error control schemes, such as Automatic Repeat reQuest (ARQ) or Forward Error Correction (FEC) techniques, needs to be analyzed in light of the peculiarities of nanonetworks. For example, on the one hand, the long time needed to harvest enough energy to retransmit a packet in ARQ might render the data useless. On the other hand, the majority of FEC mechanisms are simply too complex for the expected computational capabilities of the nanomachines, and can also result into significant delay and energy consumption.

In parallel, new error control strategies have been recently suggested, such as low-weight Error Prevention Codes (EPCs). More specifically, it was shown in [9] that the reduction of the average number of logic ones transmitted per packet results in a decrease in the overall molecular-absorption noise and interference powers. However, the reduction of the coding weight requires the transmission of longer data packets, which results in a higher energy consumption both at the transmitter and the receiver when compared to that of uncoded transmission [5, 6, 15]. Thus, there is a need for a unified cross-layer error-control analysis, tailored to the peculiarities of nanonetworks both on the nano-device side and the communication side.

In this paper, a joint physical and link layer analysis of error-control strategies for nanonetworks in the THz band is presented. In particular, we develop a mathematical framework and analyze the tradeoffs between Bit Error Rate (BER), Packet Error Rate (PER), energy consumption and latency, for five different error-control strategies, namely, ARQ, for which we consider that a simple Cyclic Redundancy Check (CRC) is added for error detection at the receiver; FEC, based on the use of simple block codes such as Hamming codes; two types of EPCs, for which we consider the use of both constant and variable low-weight channel codes, and a hybrid scheme (HEPC), which combines both FEC and EPC.

Our analysis captures the peculiarities of the THz-band channel, especially the impact of molecular absorption on the signal propagation and noise. The physical layer is built on top of a recently proposed modulation scheme based on the transmission of femtosecond-long pulses spread in time [13]. At the link layer, we consider that nanomachines access the channel in an unregulated fashion as in [14], and account for the impact of multi-user interference. Our analysis also captures the impact on energy consumption and delay introduced by the process of coding and decoding even when utilizing simple error detection and error correction codes. In addition, the impact of energy-harvesting systems on the performance of the different error-control strategies is also taken into account. The mathematical models utilized in our framework have been validated by means of electromagnetic simulations with COMSOL Multi-physics [7]. The developed framework is then utilized to provide a fair comparison of the aforementioned error control techniques.

The remainder of this paper is organized as follows. In Sect. 2, we describe the different error-control techniques for nanonetworks and motivate the need for the developed joint physical and link layer analysis. In Sect. 3, we present our mathematical framework. In particular, we provide formulations for the BER, PER, energy consumption and latency, and we investigate the impact of energy harvesting and computational cost of each solution on the different performance metrics. In Sect. 4, we numerically investigate the performance of the different error control schemes by utilizing the developed framework and we conclude the paper in Sect. 5.

## 2 Motivation and system model

Although there have been many studies on the performance of error-control techniques for all sorts of wireless communication networks, these are not valid for nanonetworks. For example, in [19], a cross-layer analysis of error control in micro/macro wireless sensor networks is conducted, by taking into account the interdependencies between the physical, link and network layers. However, this framework cannot be used for nanonetworks because it does not capture the behavior of the THz-band channel, the unique physical layer of nanonetworks, or need for energy harvesting systems in nanomachines. Similar concerns arise with the framework in [8], which is tailored to underwater acoustic communication networks.

In this paper, we consider the following error-control techniques, whose performance drastically depends on the peculiarities of nanonetworks:

- *Automatic repeat request:* When using ARQ-based techniques, the recovery from errors relies on the retransmission of the failed packets. In the context of nanonetworks, ARQ-based techniques are advantageous because they offer minimal complexity. However, in case of frequent channel errors and, thus, retransmissions, ARQ-based schemes suffer from significant delay due to significant amount of time needed to harvest energy for a new transmission [11]. In our analysis, we will consider that a CRC field is added to each transmitted packet for error detection at the receiver. This CRC can be easily computed with a combination of logic gates and registers.
- *Forward error correction:* FEC-based techniques rely on the transmission of redundant bits within each packet to help the receiver to recover the original bits even in the presence of a few channel errors. In the context of nanonetworks, only very simple error-correcting codes should be considered, mainly due to the very limited computational capabilities of nanodevices. Otherwise, the coding/decoding delay as well as the energy spent by the processor can significantly affect the throughput. On the positive side, however, for a target BER, FEC-based techniques allow the system to properly operate with a lower signal-to-noise ratio (SNR) than ARQ-based techniques, and, thus, increase the communication distance [19]. In this study we will consider only very simple Hamming codes, which can also be implemented by means of a combination of logic gates and registers.
- *Error prevention codes:* The objective of EPC-based techniques is to prevent channel errors beforehand. This is achieved in nanonetworks by utilizing low-weight codes [9], which can effectively mitigate both the molecular absorption noise at THz-band frequencies and the multi-user interference when utilizing a on/off modulations. As a result, for the same transmitted power, the SNR at the receiver is larger than when using ARQ or FEC-type schemes. This can be leveraged to reduce the transmission power and, thus, save energy or even reduce the packet latency or, as for FEC, to increase the transmission distance. Nevertheless, the reduction of the coding weight results into the transmission of longer codewords. While for the transmitter this might not be a significant problem, mainly because the number of transmitted pulses will be expectedly low, it might become a problem for the receiver, which consumes the same energy independently of the type of symbol received. In our analysis, we will consider two types of EPCs, namely, constant low-weight codes, as in [9] and bounded low-weight codes such as in [6].
- *Hybrid error prevention codes:* Traditional hybrid error-control schemes combine ARQ with FEC schemes to improve the error resilience. In our case, however, given the impact of retransmissions, we focus on combining the benefits of FEC and EPC instead. This is achieved by designing a code that exhibits low-weight but at the same time can guarantee a target Hamming distance [15]. Ultimately, the idea is to jointly exploit the higher SNR provided by EPC with the lower SNR required by FEC to minimize the energy consumption, in such a way that the energy harvesting system does not become the bottleneck of the system. This will be achieved at the cost of delay, as we will discuss in Sect. 3.

## 3 Error-control analysis

### 3.1 Bit error rate

In this section, we derive the expressions for the BER for nanosensors. We consider that nanosensors communicate by using an on-off keying modulation spread in time (TS-OOK) [13]. Under this scheme, logic “1”s are transmitted as very short pulses, just one hundred femtoseconds long, whereas logic “0”s are transmitted as silence. Under this scheme, only one bit is transmitted per symbol and, thus, we use the terms symbol and bit interchangeably.

The transmitted signals are attenuated and distorted by the THz-band channel and, in addition, they suffer from molecular absorption noise and multi-user interference. More specifically, the signal power at the receiver, $P_{r|x}$, conditioned to the transmission of symbol $x \in [0, 1]$ is given by

$$P_{r|x}(d) = \int_B S_x(f)|H_c(f,d)|^2|H_r(f)|^2 df, \tag{1}$$

where $d$ is the distance between the transmitter and the receiver, $S_x$ is the single-sided power spectral density (p.s.d) of the transmitted signal corresponding to symbol $x$, $B$ stands for its bandwidth and $f$ refers to frequency. $H_c$ refers to the THz-band channel frequency response, which is given by

$$H_c(f,d) = \left(\frac{c}{4\pi fd}\right)\exp\left(-\frac{k_{abs}(f)d}{2}\right), \tag{2}$$

where $c$ refers to the speed of light and $k_{abs}$ is the molecular absorption coefficient of the medium. This parameter depends on the molecular composition of the transmission medium, i.e., the type and concentration of molecules found in the channel, and is computed as in [10]. $H_r$ in (1) refers to the receiver frequency response, which we consider an ideal low-pass filter with bandwidth $B$, for the time being.

The molecular absorption noise at the receiver, $\mathcal{N}_0$, is additive, Gaussian, colored and correlated to the transmitted signal [13]. The noise probability density function (p.d.f.), $f_{\mathcal{N},}$, conditioned to the transmission of symbol $x \in [0,1]$ is given by

$$f_{\mathcal{N},}(n_0|X=x) = \frac{1}{\sqrt{2\pi N_x}}e^{-\frac{1n^2}{2N_x}}, \tag{3}$$

where $n_0$ refers to noise and $N_x$ stands for the molecular absorption noise power when symbol $x$ is transmitted, which is obtained as

$$N_x(d) = \int_B S_{N_x}(f,d)|H_r(f)|^2 df. \tag{4}$$

$S_{N_x}$ is the total molecular absorption noise p.s.d. given the transmission of symbol $x$. This noise is contributed by the background atmospheric noise p.s.d., $S_{N^B}$, and the self-induced noise p.s.d., $S_{N_x^I}$, which are defined as

$$S_{N_x}(f,d) = S_{N^B}(f) + S_{N_x^I}(f,d), \tag{5}$$

$$S_{N^B}(f) = \lim_{d\to\infty} k_B T_0(1-\exp(-k_{abs}(f)d))\left(\frac{c}{\sqrt{4\pi}f_0}\right)^2, \tag{6}$$

$$S_{N_x^I}(f,d) = S_x(f)(1-\exp(-k_{abs}(f)d))\left(\frac{c}{4\pi df_0}\right)^2, \tag{7}$$

where $k_B$ is the Boltzmann constant, $T_0$ is the room temperature, $f_0$ is the design center frequency, and the rest of the terms are defined as in (1). Note that only when a non-zero signal is transmitted, e.g., a pulse, the self-induced noise is non-zero. In other words, pulses suffer from higher noise.

From [13], the p.d.f. of the multi-user interference power at the receiver, $f_I$, created by a Poisson field of interferers that transmit in an unregulated fashion, is given by

$$f_I(i) = \frac{1}{\pi i}\sum_{j=1}^{\infty}\frac{\Gamma(\gamma j+1)}{j!}\left(\frac{\pi\lambda'\alpha\Gamma(1-\gamma)}{i^\gamma}\right)^j \sin j\pi(1-\gamma), \tag{8}$$

where $i$ refers to interference power, $\Gamma(\cdot)$ stands for the gamma function, and $\gamma$ and $\alpha$ are two constants related to the channel path loss and are approximately equal to 0.95 and $1.39 \cdot 10^{-18}$, respectively, at THz frequencies [10]. $\lambda'$ is the spatial Poisson point parameter given by

$$\lambda' = \lambda_T(2T_p/T_s)p_X(X=1), \tag{9}$$

where $\lambda_T$ refers to the density of active nodes, $T_p$ refers to the symbol length, $T_s$ stands for the time between symbols, and $p_X(X=1)$ refers to the probability of a nano-device to transmit a pulse. This expression emphasizes the fact that the transmission of "silence" does not create interference to other ongoing transmissions.

We can now write the p.d.f. of the channel output $Y$ conditioned to the transmission of symbol $x \in [0,1]$ as

$$\begin{aligned} f_Y(y|X=x) = \delta(y-a_{r|x}) * f_{\mathcal{N},}(n_0 = y|X=x) \\ * \left(2yf_I(i=y^2)\right), \end{aligned} \tag{10}$$

where $\delta$ stands for the Dirac delta function, $a_{r|x}$ stands for the received symbol amplitude, obtained from (1), $f_{\mathcal{N},}$ is the p.d.f. of the noise given by (3), $f_I$ stands for the p.d.f. of the interference power given by (8), and $*$ denotes convolution.

As shown in [9], when considering a 1-bit hard receiver based on power detection, the system becomes a Binary Asymmetric Channel (BAC) and is fully characterized by the four transition probabilities:

$$\begin{aligned} p_Y(Y=0|X=0) &= \int_{th_1}^{th_2} f_Y(y|X=0)dy, \\ p_Y(Y=1|X=0) &= 1 - p_Y(Y=0|X=0), \\ p_Y(Y=0|X=1) &= \int_{th_1}^{th_2} f_Y(y|X=1)dy, \\ p_Y(Y=1|X=1) &= 1 - p_Y(Y=0|X=1), \end{aligned} \tag{11}$$

where $th_1$ and $th_2$ in (11) are two threshold values. Contrary to the classical symmetric additive Gaussian noise channel, in the asymmetric channel, there are two points at which $f_Y(y|X=0)$ and $f_Y(y|X=1)$ intersect. For the system without interference, $th_1$ and $th_2$ can be analytically computed from the intersection between two Gaussian distributions $\mathcal{N},(0,N_0)$ and $\mathcal{N},(a_1,N_1)$ respectively, which results in

$$th_{1,2} = \frac{a_{r|1}N_0}{N_0 - N_1} \pm \frac{\sqrt{2N_0N_1^2\log(N_1/N_0) - 2N_0^2N_1\log(N_1/N_0) + a_{r|1}^2N_0N_1}}{N_0 - N_1}, \tag{12}$$

where $a_{r|1}$ is the amplitude of the received signal given that a pulse has been transmitted and $N_0$ and $N_1$ stand for the distance dependent noise powers given by (4). For the general case with multi-user interference, the thresholds can only be numerically obtained.

Finally, the bit error rate for nanosensors is obtained as

$$\begin{aligned} BER = & \, p_Y(Y = 1|X = 0)p_X(X = 0) \\ & + p_Y(Y = 0|X = 1)p_X(X = 1). \end{aligned} \tag{13}$$

Thus, the BER depends on the probability distribution of the transmitted symbol $p_X$, both directly as well as indirectly through the noise and interference probability distributions given by (3) and (8). On its turn, the source probability distribution depends on the coding weight, as we show next.

### 3.2 Impact of the coding weight on the bit error rate

In this section, we derive the relation between the coding weight, the transmitted symbol probability distribution and the BER for the different error control techniques under analysis.

We consider that the source at the transmitter generates $k$-bit-long constant-length messages. The total number of possible $k$-bit messages is given by $K = 2^k$. We also consider that the messages are equiprobable and, thus, the probability to transmit a given message is $p = 1/K$. The weight of a message, defined as the number of bits equal to "1", is denoted by $w$. For a given $k$, the total number of messages $\mathcal{W}$ with weight exactly equal to $w$ is given by the binomial coefficient:

$$\mathcal{W}(k, w) = \binom{k}{w} = \frac{k!}{(k - w)!w!}. \tag{14}$$

When the message is transmitted without any further coding, such as in ARQ, it is easy to show that the message average weight $\tilde{w}^{ARQ}$ is equal to

$$\tilde{w}^{ARQ} = \frac{1}{K}\sum_{w=0}^{k} w\mathcal{W}(k, w) = \frac{1}{K}\sum_{w=0}^{k}\frac{k!}{(k - w)!(w - 1)!} = \frac{k}{2}. \tag{15}$$

Therefore, the probability to transmit symbol $x \in [0, 1]$ for ARQ is given by

$$p_X^{ARQ}(X = 1) = p_X^{ARQ}(X = 0) = \frac{1}{2}. \tag{16}$$

Let us now consider that each message is encoded with $n$ bits, $n \geq k$. In general, existing FEC schemes make use of all the possible codewords independently of their weight. As a result, it can be shown that

$$p_X^{FEC}(X = 0) = p_X^{FEC}(X = 1) = \frac{1}{2}, \tag{17}$$

which is the same as for ARQ.

For EPC, the objective is to reduce the coding weight in order to mitigate noise and interference. In our analysis, we consider two different EPC schemes, namely, EPCI and EPCII. First, we consider low constant weight codes, such as ration codes [18]. In this case, in order to be able to encode all the possible $k^{EPC}$-bit source messages into codewords with constant weight $w_c$, the following condition for the encoded message size $n^{EPCI}$ must be satisfied:

$$\mathcal{W}\left(n^{EPCI}, w_c\right) \geq 2^{k^{EPC}}. \tag{18}$$

Second, we consider the case in which a constant weight is not required, but only the maximum weight is bounded to $w_m$. Now, the necessary condition on the coded message length $n^{EPCII}$ is given by

$$\mathcal{W}\left(n^{EPCII}, 0\right) + \mathcal{W}\left(n^{EPCII}, 1\right) + \ldots + \mathcal{W}\left(n^{EPCII}, w_m\right) \geq 2^{k^{EPC}}. \tag{19}$$

Then, the probability to transmit symbol $x \in [0, 1]$ is given by

$$\begin{aligned} p_X^{EPCi}(X = 1) = & \frac{\tilde{w}^{EPCi}}{n^{EPCi}}, \\ p_X^{EPCi}(X = 0) = & 1 - \frac{\tilde{w}^{EPCi}}{n^{EPCi}}, \end{aligned} \tag{20}$$

where

$$\begin{aligned} \tilde{w}^{EPCI} & = w_c, \text{ for } i = I, \text{ and} \\ \tilde{w}^{EPCII} & = \frac{1}{2^{k^{EPC}}}\sum_{w=0}^{w_m} w\mathcal{W}\left(n^{EPCII}, w\right), \text{ for } i = II. \end{aligned} \tag{21}$$

Finally, for Hybrid EPC with $k^{HEPC}$-bit-long words, able to simultaneously keep a low constant code weight $w_c$ and a code distance $d_c$, the codeword length $n^{HEPC}$ is given by [15]

$$n^{HEPC} = \begin{cases} 2^{k^{HEPC}}\dfrac{d_c}{2} & \text{for even } d_c \\ 2^{k^{HEPC}}\left\lceil\dfrac{d_c}{2}\right\rceil - 1 & \text{for odd } d_c, \end{cases} \tag{22}$$

and the expected weight is

$$\tilde{w}^{HEPC} = \begin{cases} \dfrac{d_c}{2} & \text{for even } d_c \\ \left\lceil\dfrac{d_c}{2}\right\rceil - \dfrac{1}{2^{k^{HEPC}}} & \text{for odd } d_c, \end{cases} \tag{23}$$

where $\lceil \cdot \rceil$ is the ceiling function.

The probability to transmit symbol $x \in [0, 1]$ can be obtained from (20) with $n^{HEPC}$ and $\tilde{w}^{HEPC}$. Finally, the BER for each error control technique can be now obtained by substituting (16), (17) or (20) in (13), for ARQ, FEC and EPCI, EPCII or HEPC, respectively.

### 3.3 Packet error rate

Based on the BER, the PER for each error-control scheme can be calculated as follows. For ARQ, we consider that a simple CRC is utilized for error detection. Assuming that all the errors in a packet can be detected, the PER of a single transmission of a packet with payload $l$ bits is given by

$$PER^{ARQ} = 1 - \left(1 - BER^{ARQ}\right)^{l'}, \tag{24}$$

where $l' = l + 16$ bits because of the CRC.

For FEC, the block error rate (BLER) is given by

$$BLER^{FEC} = \sum_{j=t+1}^{n} \binom{n}{j} \left(BER^{FEC}\right)^{j} \left(1 - BER^{FEC}\right)^{(n-j)}, \tag{25}$$

where $n$ refers to the coded message size or block size and $t$ is the error correction capability of the code.

Since the packet payload $l$ can be larger than the block length $k$, the PER for FEC is finally obtained as

$$PER^{FEC} = 1 - \left(1 - BLER^{FEC}\right)^{\lceil \frac{l}{k} \rceil} \tag{26}$$

where $\lceil \frac{l}{k} \rceil$ is the number of blocks per packet payload.

Similarly, when using EPC type I or type II, the total packet length is increased from $l$ bits to $\lceil l/k^{EPC} \rceil n^{EPCi}$, $i \in$ [I,II]. The PER can then be written as

$$PER^{EPCi} = 1 - \left(1 - BER^{EPCi}\right)^{\left\lceil \frac{l}{k^{EPC}} \right\rceil n^{EPCi}}. \tag{27}$$

Finally, in the case of HEPC with distance $d_c$, the correcting capability is $t_c = \left\lfloor \frac{d_c - 1}{2} \right\rfloor$, and the BLER and PER are respectively given by

$$BLER^{HEPC} = \sum_{j=t_c+1}^{n^{HEPC}} \binom{n^{HEPC}}{j} \left(BER^{HEPC}\right)^{j} \left(1 - BER^{HEPC}\right)^{\left(n^{HEPC} - j\right)}, \tag{28}$$

and

$$PER^{HEPC} = 1 - \left(1 - BLER^{HEPC}(n, t)\right)^{\left\lceil \frac{l}{k^{HEPC}} \right\rceil}. \tag{29}$$

Starting from the packet error rate, we can now estimate the energy consumption and packet latency for the error control strategies under study.

### 3.4 Energy consumption

The energy consumption associated to each error control technique is mainly determined by the energy required per transmission and the expected number of retransmissions needed to complete the data transaction. In this section, we focus on the energy consumption due to communication. In Sect. 3.6, we will compute the additional energy needed to perform the associated computational tasks.

The total energy $E$ consumed in the successful transmission and reception of a packet is given by

$$E = E_{TX} + E_{RX}, \tag{30}$$

where $E_{TX}$ is the energy consumed at the transmitter and $E_{RX}$ is the energy consumed at the receiver.

To estimate these two terms, we proceed as follows. In our analysis, we consider that nanomachines communicate in an uncoordinated fashion and without any sort of initial handshake or channel reservation mechanism. There are several reasons for this. First of all, when utilizing a pulse-based modulation such as TS-OOK, there is no carrier to sense and the detection of individual pulses is challenging due to their very short duration. Even if possible, the detection of the pulses on the transmitter side does not provide relevant information about the channel status at the receiver. Similarly, the very short duration of the transmitted pulses creates almost virtual orthogonal channels for nanomachines. Still, interference might occur and we capture that in our model as described in Sect. 3.1. Finally, as we showed in [14], an initial handshake is only needed in case that the nanomachines are likely to be unavailable, mainly because they deplete their batteries. However, in this work, we focus on energy-harvesting nanonetworks with perpetual operation and design the network in a way that such probability is minimal. In Sect. 3.7, we formulate the conditions to guarantee the perpetual operation of the network.

Therefore, in order to successfully transmit a data packet, a nanomachine needs to complete a two-way DATA-ACK process for ARQ, and just one-way DATA transaction for FEC, EPCI/II, and HEPC. The energy consumption at the transmitter for ARQ is thus given by

$$E_{TX}^{ARQ} = n_{ret}^{ARQ} \left(E_{tx}^{D-ARQ} + E^{CRC} + p_{s}^{D-ARQ} p_{s}^{A-ARQ} E_{rx}^{A-ARQ}\right), \tag{31}$$

where $n_{ret}^{ARQ}$ refers to the expected number of retransmissions, and is given by

$$n_{ret}^{ARQ} = \left(p_{s}^{D-ARQ} p_{s}^{A-ARQ}\right)^{-1}, \tag{32}$$

where $p_{s}^{D-ARQ}$ and $p_{s}^{A-ARQ}$ refer to the probability of successfully receiving a DATA packet and ACK packet, respectively, and can be obtained from (24), with packet length $l = l^{DATA}$ bits and $l = l^{ACK}$ bits, respectively. Similarly, $E_{tx}^{D-ARQ}$ and $E_{rx}^{A-ARQ}$ stand for the energy to transmit a DATA packet and the energy to receive an ACK packet. For any packet with size $l$ bits and probability to transmit $X = 1$, $p_X(X = 1)$, the energy consumption in transmission and in reception can be obtained as

$$\begin{aligned} E_{tx}(l, \tilde{w}) &= lE_{tx-p}p_X(X = 1), \\ E_{rx}(l) &= lE_{rx-p}, \end{aligned} \tag{33}$$

where $E_{tx-p}$ and $E_{rx-p}$ correspond to the energy to transmit and detect a pulse, respectively. $E^{CRC}$ in (31) refers to the energy consumed in computing the CRC field of the DATA packet, and its derivation is given in Sect. 3.6.

Similarly, the energy consumption for EC = FEC, EPCI/II or HEPC is given by

$$E_{TX}^{EC} = n_{ret}^{EC}\left(E_{tx}^{D-EC} + E_{code}^{D-EC}\right), \tag{34}$$

where the number of retransmissions is given by

$$n_{ret}^{EC} = \left(p_s^{D-EC}\right)^{-1}. \tag{35}$$

$p_s^{D-EC}$ is obtained for each error control technique from (26), (27), (29), respectively. $E_{tx}^{D-EC}$ in (34) can be obtained from (33) with the corresponding total number of bits transmitted with each technique (Sect. 3.3). $E_{code}^{D-EC}$ in (34) refers to the energy consumed to code the DATA packet, and its derivation is given in Sect. 3.6.

In a similar way, the energy consumption at the receiver, $E_{RX}$ in (30), for ARQ is given by

$$E_{RX}^{ARQ} = n_{ret}^{ARQ}\left(E_{rx}^{D-ARQ} + E^{CRC} + E_{tx}^{A-ARQ}\right), \tag{36}$$

and for EC = FEC, EPC-I/II and HEPC can be obtained as

$$E_{RX}^{EC} = n_{ret}^{EC}\left(E_{rx}^{D-EC} + E_{decode}^{D-EC}\right). \tag{37}$$

At this point, all the elements to compute the energy consumption for all the error control techniques has been defined.

### 3.5 Latency

The computation of the packet latency follows a very similar approach to that of the energy consumption. In particular, the packet latency for ARQ is given by

$$\begin{aligned} T^{ARQ} = n_{ret}^{ARQ}\big(&T_{tx}^{D-ARQ} + T^{CRC} \\ &+ p_s^{D-ARQ}p_s^{A-ARQ}\left(2T_{prop} + T^{CRC} + T_{tx}^{A-ARQ}\right) \\ &+ \left(1 - p_s^{D-ARQ}p_s^{A-ARQ}\right)T_{t/o}\big), \end{aligned} \tag{38}$$

where $T_{tx}^{D-ARQ}$ and $T_{tx}^{A-ARQ}$ are the DATA transmission time and ACK transmission time, and can be directly obtained from the physical-layer data-rate and the DATA and ACK lengths, respectively. $T^{CRC}$ in (38) refers to the latency introduced by computing the CRC field of the DATA packet, and its derivation is given in Sect. 3.6. $T_{prop}$ is the propagation time and $T_{t/o}$ refers to the time-out before retransmission. The number of retransmissions $n_{ret}^{ARQ}$ and the probabilities $p_s^{D-ARQ}$ and $p_s^{A-ARQ}$ are computed as before.

Similarly, for EC=FEC, EPCI/II or HEPC, the packet latency is given by

$$\begin{aligned} T^{EC} = n_{ret}^{EC}\big(&T_{tx}^{D-EC} + T_{code}^{EC} + p_s^{D-EC}\left(T_{prop} + T_{decode}^{EC}\right) \\ &+ \left(1 - p_s^{D-EC}\right)T_{t/o}\big), \end{aligned} \tag{39}$$

where the transmission time $T_{tx}^{D-EC}$ is computed according to the physical layer data rate and the total number of bits to be transmitted with each error control technique EC. $T_{code}^{EC}$ and $T_{decode}^{EC}$ refer to the latency introduced by coding and decoding the DATA packet, respectively, and their derivation is given in the next section for the different error control techniques.

### 3.6 Impact of coding & decoding on energy and latency

The limited computational capabilities of nanomachines motivate the utilization of only very simple error control strategies, such as a 16-bit CRC for error detection in ARQ, a Hamming (15,11) for single-bit error correction in FEC, and 16-bit low-weight codes for EPCI, EPCII, and HEPC. All these techniques can be implemented in hardware by a combination of logic gates and hold and shift registers [16]. This makes them very attractive for nanonetworks based on resource-limited nanomachines. In this section, we estimate the energy and latency introduced by each error control technique.

A CRC can be easily implemented in hardware with a set of shift registers and a combination of exclusive OR (XOR) gates. In particular, for the standardized CRC-16 with polynomial generator given by $x^{16} + x^{15} + x^2 + 1$, 16 single-bit registers and 3 OR gates are needed. The computation of a CRC for an $l$-bits-long packet requires a total of $l$ shift cycles. By considering the energy consumption by an *xor* gate to be negligible, the energy, $E^{CRC}$, consumed to compute a 16-bit CRC is obtained as

$$E^{CRC} = 16l\left(E_{shift} + E_{hold}\right), \tag{40}$$

where $E_{shift}$ and $E_{hold}$ stand for the energy consumed to shift and hold the registry value, respectively. Finally, we need to take into account that the CRC is performed both at the transmitter and the receiver.

Similarly, the latency introduced by the computation of a 16-bit CRC is given by

$$T^{CRC} = lT_{cycle}, \tag{41}$$

where $T_{cycle}$ is the inverse of the clock at the nanomachine.

Hamming codes can also be implemented with a combination of parallel load and shift registers as well as XOR and AND gates. In particular, for a Hamming (15,11), the energy $E_{HC}$ consumed to encode $k = 11$ bits into $n = 15$ bits is given by,

$$E^{HC}_{code} = n(E_{load} + E_{hold}), \tag{42}$$

where $E_{load}$ and $E_{hold}$ stand for the energy consumed to load and hold the registry value, respectively. As before, we consider the energy consumed by the AND and XOR gates, needed to compute each one of the parity bits in the codeword, to be negligible. Note that this process needs to be repeated for a total of $\lceil l/k \rceil$ blocks. Then, the coding latency introduced per block is given by,

$$T^{HC}_{code} = 2T_{cycle}, \tag{43}$$

where we consider that the registers can be load in parallel.

In terms of decoding, the energy $E^{HC}_{decode}$ consumed in decoding the $n$-bit-long block can be computed as

$$E^{HC}_{decode} = n(E_{load} + E_{hold}) + n^2\left(E_{shift} + E_{hold}\right), \tag{44}$$

and the latency introduced in the decoding process is

$$T^{HC}_{decode} = (n+1)T_{cycle}, \tag{45}$$

where we take into account that first the $n$ bits are loaded in parallel (at once) in $n$ registers and, then, the bits are shifted one-by-one in $n$ consecutive cycles.

For EPC codes, both type I (constant low weight) and type II (max-bound low weight), a combination of logic gates and a total of $k^{EPCi} + n^{EPCi}$ parallel-load registers are needed, both to code and decode a block. In this case, the energy $E^{EPCi}$ consumed for coding/decoding is given by

$$E^{EPCi} = \left(k^{EPCi} + n^{EPCi}\right)(E_{load} + E_{hold}). \tag{46}$$

This is significantly less energy than that needed for FEC. Similarly, the latency $T^{EPCi}$ introduced to decode the received codeword is

$$T^{EPCi} = 2T_{cycle}. \tag{47}$$

As for FEC, this process needs to be repeated for a total of $\lceil l/k^{EPCi} \rceil$ blocks.

Finally, for HEPC, the coding/decoding energy and latency are given by (42), (43), (44), and (45), respectively, where $n$ should be replaced by $n^{HEPC}$ given by (22).

### 3.7 Impact of energy harvesting on latency

Nanosensors require energy harvesting systems to replenish their batteries. Amongst others, one of the main alternatives is to use piezoelectric nano-generators [21], which convert vibrational and kinetic energy into electricity by exploiting the piezoelectric behavior of Zinc Oxide nanowires. Every time that the ZnO nanowires are compressed or released, a small electric current is generated. This can be used to recharge an ultra-nano-capacitor after proper rectification. Our starting point for the analysis of the impact of energy harvesting on the latency is the model introduced in [11], which can accurately reproduce experimental measurements.

We are interested in the energy harvesting rate, i.e., the speed at which the battery is replenished, $\lambda_{harv}$. The energy in battery can be written as

$$E_{batt} = \frac{1}{2}V_g^2 C_{cap}\left(1 - \exp\left(-\frac{\Delta Q}{V_g C_{cap}} n_{cycle}\right)\right), \tag{48}$$

where $V_g$ is the generator voltage, $C_{cap}$ refers to the ultra-nano-capacitor capacitance, and $\Delta Q$ is the electric charge harvested per cycle. From this, the energy harvesting rate is obtained as:

$$\begin{aligned}\lambda_{harv} &= \frac{\partial E_{batt}}{\partial n_{cycle}} \lambda_{cycle} \\ &= \frac{1}{2} C_{cap} V_g^2 \left(2\frac{\Delta Q}{V_g C_{cap}} \exp\left(-\frac{\Delta Q}{V_g C_{cap}} n_{cycle}\right)\right. \\ &\left. -2\frac{\Delta Q}{V_g C_{cap}} \exp\left(-2\frac{\Delta Q}{V_g C_{cap}} n_{cycle}\right)\right)\lambda_{cycle},\end{aligned} \tag{49}$$

where $\lambda_{cycle}$ is the vibration frequency or compression-release rate of the ZnO nanowires, and the rest of parameters have already been defined.

Similarly as in [20], for the network to operate perpetually and uninterruptedly, the energy consumed during the transmission and reception of a packet (and all its potential retransmissions) needs to be at most equal to the energy harvested between the generation of new packets, i.e.,

$$\begin{aligned} E^{ARQ/EC}_{TX} &\le \lambda_{harv} T_{new-TX}, \\ E^{ARQ/EC}_{RX} &\le \lambda_{harv} T_{new-RX}, \end{aligned} \tag{50}$$

where $T_{new-TX}$ and $T_{new-RX}$ refer to the average time between new transmissions and new receptions, respectively. From Sects. 3.4 and 3.5, it is clear that $E^{ARQ/EC}_{TX}$ and $E^{ARQ/EC}_{RX}$ depend on the specific error control strategy. Therefore, the maximum new packet generation rate also depends and varies for the different error control techniques in our analysis. Ultimately, the average throughput $S$ that nanomachines can achieve also changes with the different error control techniques, and can be calculated as

$$S = \frac{l^{DATA}}{\max\{T_{new-TX}, T_{new-RX}\}}, \tag{51}$$

where $l^{DATA}$ refers to the actual data bits per packet.

## 4 Numerical results

In this section, we numerically compare the performance of the different error control techniques under study, in terms of BER, PER, energy consumption, latency and maximum throughput under the energy-harvesting constraint.

In our numerical analysis, we use the following parameter values. Pulses in TS-OOK are modeled as the first time derivative of a one-hundred-femtosecond long Gaussian pulse with a total energy of 1 aJ (which corresponds to a peak power of approximately 1 $\mu$W). The pulse duration is $T_p = 100\,\mathrm{fs}$, and the symbol duration is $T_s = 10\,\mathrm{ps}$. Unless the contrary is stated, the density of active nodes $\lambda_T$ in (9) is equal to 1 node/mm$^2$. The THz band channel is modeled as in [10], for a standard gaseous medium with 10% of water vapor molecules. The THz band channel model, including path-loss and molecular absorption noise, the propagation of femtosecond-long pulses, and the impact of multi-user interference models have been previously validated by means of frequency and time-domain electromagnetic simulations with COMSOL Multi-physics [7].

For the computation of the energy consumption and latency introduced by computation of the CRC or coding and decoding with the different techniques, we consider that $E_{load} = E_{hold} = E_{shift} = 0.1\,\mathrm{aJ}$, and $T_{cycle} = 1\,\mathrm{ps}$. These values have been estimated from the expected capabilities and constraints of nanoscale processors, first discussed in [2].

The piezoelectric energy harvesting system has the following parameters as in [11]. We consider a capacitor with $C_{cap} = 9\,\mathrm{nF}$ charged at $V_g = 0.42\,hboxV$ for the computation of the energy in the nano battery (48). For the computation of the energy harvesting rate $\lambda_{harv}$ in (49), an ambient vibration with an average time between vibrations $\lambda_{cycle} = 50$ cycles/s is considered. The charge $\Delta Q$ harvested per cycle is 6 pC.

### 4.1 Bit and packet error rate

The BER, given by (13), for the different error control techniques is shown in Fig. 1 as a function of the distance between the transmitter and the receiver. Contrary to classical wireless networks, in which the BER only depends on the modulation and SNR at the receiver, in nanonetworks, different error control techniques result in different BER. This is a result of the relation between the different error control techniques and the coding weight. The word length $k$, codeword size $n$, average coding weight $\tilde{w}$ and resulting probability to transmit a pulse $p_X(X = 1)$ for each coding technique are summarized in Table 1. Clearly, for HEPC with minimum coding weight, the BER before decoding is the lowest. On the contrary, for both ARQ and FEC, which transmit 0s and 1s with the same probability, the BER before decoding is almost one order of magnitude higher than for HEPC. The different variations of EPC codes range in between these two bounds.

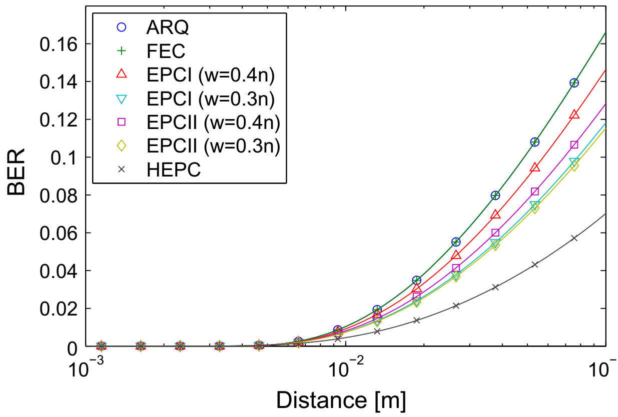

**Fig. 1** BER for different error control techniques

Similarly, in Fig. 2, the PER for ARQ, FEC, EPCs and HEPC, given respectively by (24), (26), (27) and (29), is shown as a function of the transmission distance. There are several observations to be made. First, focusing only the ARQ and FEC, the behavior is as expected: the former exhibits the worst PER, whereas the latter is able to significantly increase the transmission distance, as a result of what is commonly referred to as *hop extension*. This improvement in terms of PER could be also used to reduce the transmission power and, thus, the energy consumption. However, adaptive transmission power control is out of the capabilities of nanomachines. Second, despite HEPC is able to minimize the BER before decoding, the need to transmit very long codewords to minimize the average coding weight results into a very high PER, even if the HEPC we use can correct all one-bit errors. Finally, EPCs on their turn are able to minimize the PER beyond that of FEC. This is inline with our results in [9], where we showed there is an optimal coding weight for which the system performance is maximized, different from the minimal coding weight.

### 4.2 Energy consumption

The energy consumption in transmission and in reception, $E_{TX}$ and $E_{RX}$, respectively, are shown in Fig. 3 for the different error control techniques and as a function of the distance between the transmitter and the receiver. For the computation of $E$ we have taken into account both the communication energy consumption, given in Sect. 3.4, and the computation energy consumption, given in Sect. 3.6. There are several observations to be made. First,

**Table 1** Summary of the parameters associated to each error control technique

| | ARQ<br>CRC-16 | FEC<br>Hamm (15,11) | EPCI<br>$w_c = 0.4n$ | EPCI<br>$w_c = 0.3n$ | EPCII<br>$w_m = 0.4n$ | EPCII<br>$w_m = 0.3n$ | HEPC<br>$d_c = 3$ |
|---|---|---|---|---|---|---|---|
| Block size $k$ | 1024 | 11 | 16 | 16 | 16 | 16 | 4 |
| Number of blocks | 1 | 94 | 64 | 64 | 64 | 64 | 256 |
| Codeword size $n$ | 1024 | 15 | 20 | 22 | 19 | 21 | 31 |
| Average weight $\tilde{w}$ | 512 | 7.5 | 8 | 6 | 5.98 | 5.43 | 1.94 |
| Pulse probability | 0.5 | 0.5 | 0.4 | 0.27 | 0.31 | 0.25 | 0.06 |
| Total num of bits $l$ | 1040 | 1410 | 1280 | 1408 | 1216 | 1344 | 7936 |
| Energy $E_{code}$ (fJ) | 3.28 | 0.28 | 0.46 | 0.49 | 0.45 | 0.47 | 1.59 |
| Energy $E_{decode}$ (fJ) | 3.28 | 4.23 | 0.46 | 0.49 | 0.45 | 0.47 | 49.20 |
| Latency $T_{code}$ (ns) | 1.02 | 0.19 | 0.13 | 0.13 | 0.13 | 0.13 | 0.51 |
| Latency $T_{decode}$ (ns) | 1.02 | 1.50 | 0.13 | 0.13 | 0.13 | 0.13 | 8.19 |

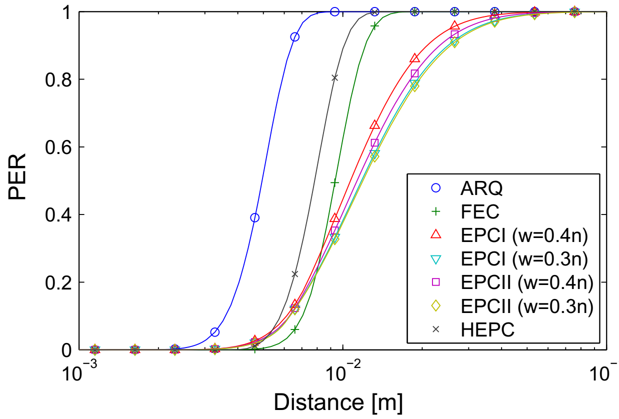

**Fig. 2** PER for different error control techniques

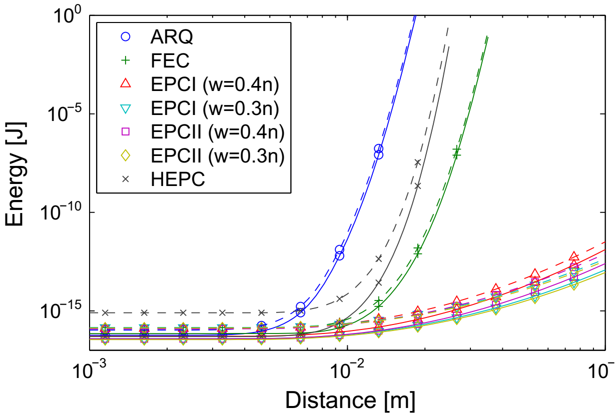

**Fig. 3** Energy consumption per packet at the transmitter (*solid lines*) and at the receiver (*dashed lines*) for different error control techniques

the energy consumption at the receiver (denoted by the dashed lines) is significantly higher than the energy consumption at the transmitter, mainly because the receiver does not benefit from the reduction of the coding weight in its communication energy consumption (33), and, moreover, the decoding process is usually much more energy demanding. Second, the lower PER achieved by EPCs allows them to reduce the number of retransmissions and this results into clear energy savings. This will be especially beneficial when taking into account the harvesting limitations below.

A summary of the coding/decoding energy consumption is given in Table 1. Despite EPCs require more energy to encode a codeword than FEC, the much lower decoding energy allows EPCs to achieve overall energy savings. The decoding of HEPC requires a very large amount of energy, mainly because of the very long codewords needed to reduce the coding weight. It is also relevant to note that the hardware implementation of a CRC-16 field results in non-negligible energy consumption both at the transmitter and the receiver.

### 4.3 Latency

In Fig. 4, the packet latency $T$, which is given by (38) and (39) for ARQ and the other error control mechanisms, respectively, is shown as a function of the transmission distance. The trend is very similar to that of the energy explained above, and it is mainly governed by the large number of retransmissions needed to successfully transmit a packet. In Table 1, the latency due to coding/decoding introduced by each technique is presented. It is relevant to note that the delay introduced by ARQ is almost one order magnitude larger than that introduced by EPCs. When it comes to FEC, the delay introduced in the coding process is almost negligible when compared to the decoding delay. HEPC suffers from very long codewords, which result into very longs delay especially in terms of decoding.

### 4.4 Throughput

As discussed in Sect. 3.7, the need for nanomachines to harvest energy to operate results into a major constraint on

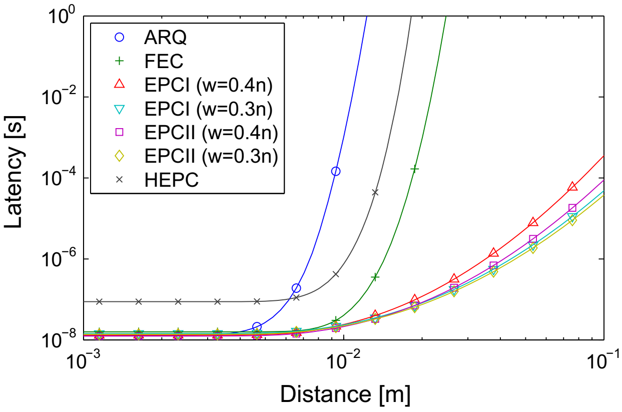

**Fig. 4** Latency per packet for different error control techniques

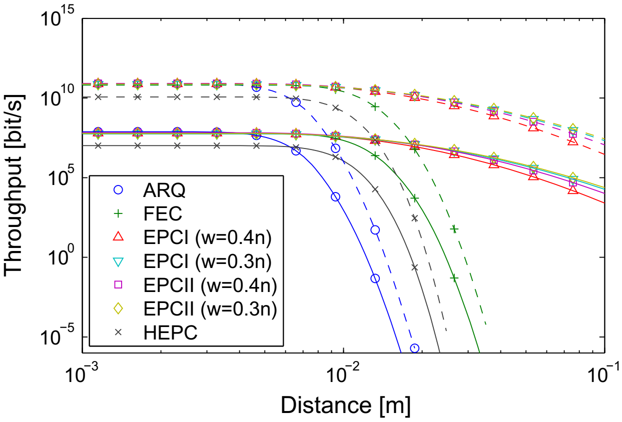

**Fig. 5** Average throughput for different error control techniques with energy constraints (*solid lines*) and without energy constraints (*dashed lines*)

the achievable throughput. In Fig. 5, the throughput $S$ given by (51), is illustrated as a function of the distance, for the different error control techniques under study (solid lines). For illustration purposes, we plot in the same figure the achievable throughput without the energy-harvesting constraints (dashed line). As in terms of energy consumption and latency, EPCs can maximize the throughput under the energy-harvesting constraint. Still, the very low rate at which the energy harvester operates results into a reduction of the throughput by almost three orders of magnitude.

## 5 Conclusions

Nanonetworks will boost the applications of nanotechnology in many fields of our society, ranging from healthcare to homeland security and environmental protection. However, there are many challenges in enabling the communication in nanonetworks. We acknowledge that there is still a long way to go before having autonomous nanomachines, but we believe that hardware-oriented research and communication-focused investigations will benefit from being conducted in parallel from an early stage.

In this paper, we have performed a joint physical and link layer analysis of error control techniques for nanonetworks. In particular, we have developed a mathematical framework to investigate the tradeoffs between error correction capabilities, energy consumption and latency, for a total of five different error control schemes, which included ARQ based on a 16-bit CRC, FEC based on Hamming (15,11) codes, constant and bound low-weight EPCs, and a hybrid EPC able to minimize the coding weight while guaranteeing a minimum distance to allow error correction. The analysis has capture the peculiarities of the THz-band channel, the physical layer, the network layer, the computational resources needed to implement the aforementioned error control techniques and the need of energy harvesting systems for perpetual operation. The results show low-weight EPCs, which are designed with the behavior of molecular absorption noise and multi-user interference in mind, achieve the lower PER, specially for interference-limited scenarios. As a result, they area also able to minimize both the energy consumption and latency.

## References

1. Abadal, S., Alarcon, E., Cabellos-Aparicio, A., Lemme, M., & Nemirovsky, M. (2013). Graphene-enabled wireless communication for massive multicore architectures. *IEEE Communications Magazine*, *51*(11), 137–143.
2. Akyildiz, I. F., & Jornet, J. M. (2010). Electromagnetic wireless nanosensor networks. *Nano Communication Networks (Elsevier) Journal*, *1*(1), 3–19.
3. Bai, P., Zhu, G., Liu, Y., Chen, J., Jing, Q., Yang, W., et al. (2013). Cylindrical rotating triboelectric nanogenerator. *ACS Nano*, *7*(7), 6361–6366.
4. Cabellos-Aparicio, A., Llatser, I., Alarcon, E., Hsu, A., & Palacios, T. (2015). Use of thz photoconductive sources to characterize tunable graphene rf plasmonic antennas. *IEEE Transactions on Nanotechnology*, *14*(2), 390–396.
5. Chi, K., Zhu, Y. H., Jiang, X., & Leung, V. (2014). Energy-efficient prefix-free codes for wireless nano-sensor networks using ook modulation. *IEEE Transactions on Wireless Communications*, *13*(5), 2670–2682.
6. Chi, K., Zhu, Y. H., Jiang, X., & Tian, X. (2013). Optimal coding for transmission energy minimization in wireless nanosensor networks. *Nano Communication Networks (Elsevier) Journal*, *4*(3), 120–130.
7. COMSOL Multiphysics Simulation Software: COMSOL. http://www.comsol.com/products/multiphysics/.
8. Domingo, M. C., & Vuran, M. C. (2012). Cross-layer analysis of error control in underwater wireless sensor networks. *Computer Communications (Elsevier) Journal*, *35*(17), 2162–2172.

9. Jornet, J. M. (2014). Low-weight error-prevention codes for electromagnetic nanonetworks in the terahertz band. *Nano Communication Networks (Elsevier) Journal*, *5*(1–2), 35–44.
10. Jornet, J. M., & Akyildiz, I. F. (2011). Channel modeling and capacity analysis of electromagnetic wireless nanonetworks in the terahertz band. *IEEE Transactions on Wireless Communications*, *10*(10), 3211–3221.
11. Jornet, J. M., & Akyildiz, I. F. (2012). Joint energy harvesting and communication analysis for perpetual wireless nanosensor networks in the terahertz band. *IEEE Transactions on Nanotechnology*, *11*(3), 570–580.
12. Jornet, J. M., & Akyildiz, I. F. (2013). Graphene-based plasmonic nano-antenna for terahertz band communication in nanonetworks. *IEEE JSAC, Special Issue on Emerging Technologies for Communications*, *12*(12), 685–694.
13. Jornet, J. M., & Akyildiz, I. F. (2014). Femtosecond-long pulse-based modulation for terahertz band communication in nanonetworks. *IEEE Transactions on Communications*, *62*(5), 1742–1754.
14. Jornet, J. M., Pujol, J. C., & Pareta, J. S. (2012). Phlame: A physical layer aware mac protocol for electromagnetic nanonetworks in the terahertz band. *Nano Communication Networks (Elsevier) Journal*, *3*(1), 74–81.
15. Kocaoglu, M., & Akan, O. B. (2013). Minimum energy channel codes for nanoscale wireless communications. *IEEE Transactions on Wireless Communications*, *12*(4), 1492–1500.
16. Lin, S., & Costello, D. J. (2004). *Error control coding: Fundamentals and applications* (Vol. 114). Englewood Cliffs: Pearson-Prentice Hall.
17. Priebe, S., & Kurner, T. (2013). Stochastic modeling of thz indoor radio channels. *IEEE Transactions on Wireless Communications*, *12*(9), 4445–4455.
18. Tabor, J. (1990). *Noise reduction using low weight and constant weight coding techniques*. Tech. rep., MIT, Cambridge, MA.
19. Vuran, M. C., & Akyildiz, I. F. (2009). Error control in wireless sensor networks: A cross layer analysis. *IEEE/ACM Transactions on Networking*, *17*(4), 1186–1199.
20. Wang, P., Jornet, J. M., Abbas Malik, M., Akkari, N., & Akyildiz, I. F. (2013). Energy and spectrum-aware mac protocol for perpetual wireless nanosensor networks in the terahertz band. *Ad Hoc Networks (Elsevier) Journal*, *11*(8), 2541–2555.
21. Wang, Z. L. (2008). Towards self-powered nanosystems: From nanogenerators to nanopiezotronics. *Advanced Functional Materials*, *18*(22), 3553–3567.

**Nadine Akkari** received her B.S. and the M.S. degrees in Computer Engineering from University of Balamand, Lebanon, in 1997 and 1999, respectively. She received her Master degree in Telecommunications Networks from Saint Joseph University and the Faculty of Engineering of the Lebanese University, Lebanon, in 2001 and a Ph.D. degree in Telecommunications Networks from National Superior School of Telecommunications (ENST), France, in 2006. She is currently an associate professor with the faculty of Computing and Information Technology at King Abdulaziz University, Jeddah, Saudi Arabia. She is a senior member of IEEE. Her research interests are in wireless networks, heterogeneous networks, and Nanonetworks.

**Josep Miquel Jornet** received the Engineering Degree in Telecommunication and the Master of Science in Information and Communication Technologies from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 2008. He received the Ph.D. degree in Electrical and Computer Engineering from the Georgia Institute of Technology, Atlanta, GA, in 2013, with a fellowship from “la Caixa” (2009–2010) and Fundacion Caja Madrid (2011–2012). He is currently an Assistant Professor with the Department of Electrical Engineering at the University at Buffalo, The State University of New York. From September 2007 to December 2008, he was a visiting researcher at the Massachusetts Institute of Technology (MIT), Cambridge, under the MIT Sea Grant program. He was the recipient of the Oscar P. Cleaver Award for outstanding graduate students in the School of Electrical and Computer Engineering, at the Georgia Institute of Technology in 2009. He also received the Broadband Wireless Networking Lab Researcher of the Year Award at the Georgia Institute of Technology in 2010. He is a member of the IEEE and the ACM. His current research interests are in electromagnetic nanonetworks, graphene-enabled wireless communication, Terahertz Band communication networks and the Internet of Nano-Things.

**Pu Wang** received the B.S. degree in Electrical Engineering from the Beijing Institute of Technology, Beijing, China, in 2003 and the M.Eng. degree in Computer Engineering from the Memorial University of Newfoundland, St. Johns, NL, Canada, in 2008. He received the Ph.D. degree in Electrical and Computer Engineering from the Georgia Institute of Technology, Atlanta, GA, in 2013. He is currently an Assistant Professor with the Department of Electrical Engineering and Computer Science, Wichita State University, Wichita, KS. He was named BWN Lab Researcher of the Year 2012, Georgia Institute of Technology. He received the TPC top ranked paper award of IEEE DySPAN 2011. He was also named Fellow of the School of Graduate Studies, 2008, Memorial University of Newfoundland. He is a member of IEEE. His research interests include wireless sensor networks, cognitive radio networks, software defined networks, nanonetworks, multimedia communications, wireless communications in challenged environment, and cyber-physical systems.

**Dr. Etimad Fadel** Department of Computer Science, King Abdulaziz University, Jeddah, Saudi Arabia Received her Bachelor's degree in Computer Science at King Abdul Aziz University with Senior Project title ATARES: Arabic Character Analysis and Recognition in 1994. She was awarded the Mphil/PhD degree in computer science at De Montfort University (DMU), Liecester, UK. The Thesis title was Distributed Systems Management Service in 2007. Currently she is working as an Assistant Professor at the Computer Science Department at KAU. During this time she was appointed in different administrative positions. She was the Vice-Dean of the Faculty FCIT at the Girl's section. This was between 2008 and 2010. Her main research interest is Distributed Systems, which are developed based on middleware technology. Currently she is looking into and working on Wireless Networks, Internet of Things and Internet of Nano-Things. In addition, she is working on smart grids and HetNets.

**L. Elrefaei** received her B.Sc. degree with honors in Electrical Engineering (Electronics & Telecommunications) in 1997. The M.Sc. in 2003 and Ph.D. in 2008 both in Electrical Engineering (Electronics) from Faculty of Engineering at Shoubra, Benha University, Egypt. She has held a number of faculty positions at Benha University, as Teaching Assistant from 1998 to 2003, as an Assistant Lecturer from 2003 to 2008, and as an Assistant Professor (referred to as Lecturer position in the Egyptian academic system) from 2008 to date. She is currently serving as an Assistant Professor at King Abdulaziz university, Jeddah, Saudi Arabia. She is a member of the advanced wireless Networking Group in King Abdulaziz university. She is a member of IEEE. Her research interests include wireless networks, nano networks, and computational intelligence.

**Muhammad Ghulam Abbas Malik** was born in Lahore, Pakistan, in 1979. He received his Bachelor of Science (B. Sc. Major in Applied & Pure Mathematics and Statistics) and Master of Science in Computer Science from University of the Punjab in 1999 and 2003 respectively. From 2002 to 2005, he served as Principal research and teamlead on European Union's project Punjabi Language and Transliteration Tool (PLATT) in LOK SUJAG, Lahore. In 2006, he received his Master in Linguistics and Computer Science from University of Paris 7–Denis Didrot. He received his Ph.D. degree in Computer Science from University of Grenoble (Ex. University Joseph Fourier), Grenoble, France, in July 2010, under the supervision of Prof. Christian Boitet (University of Grenoble 1, France) and Prof. Pushpak Bhattacharyya (Indian Institute of Technology Bombay, India). Currently, he is an Assistant Professor with Department of Computer Science, Faculty of Computing and Information Technology, University of Jeddah, Jeddah, Saudi Arabia. He was the recipient of Overseas Scholarship award from Higher Education Commission of Pakistan in 2005. He is member of IEEE, ACM and ACL. His current research interests are in electromagnetic nanonetworks, graphene-enabled wireless communication, Terahertz Band communication networks and the Internet of Nano-Things. He is founding member of Advanced Wireless Networking Group, King Abdulaziz University.

**Suleiman Almasri** received the Computer Science degree from Philadelphia University, Amman, Jordan in 1998 and M.S. degree in computer science from Amman Arab University, Amman, Jordan in 2005, and the Ph.D. degree in computer science (wireless and mobile networks) from Anglia Ruskin University, Chelmsford, UK in 2009. He is currently an assistant professor of computer science at King Abdulaziz University, Jeddah, KSA and a founding member of the Advanced Wireless Networking Group in the university. His research interests include wireless networks, nano networks, Terahertz band communication, optical networks, mobile computing, internet of nano-things, location based services and smart grid. His research has been supported by the King AbdulazizCity of Science and Technology (KACST), KSA.

**Ian F. Akyildiz** (M'86-SM'89-F'96) received the B.S., M.S., and Ph.D. degrees in Computer Engineering from the University of Erlangen-Nurnberg, Germany, in 1978, 1981 and 1984, respectively. Currently, he is the Ken Byers Chair Professor in Telecommunications with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, the Director of the Broadband Wireless Networking Laboratory and Chair of the Telecommunication Group at Georgia Tech. Dr. Akyildiz is an honorary professor with the School of Electrical Engineering at Universitat Politecnica de Catalunya (UPC) in Barcelona, Catalunya, Spain and founded the N3Cat (NaNoNetworking Center in Catalunya). Since September 2012, Dr. Akyildiz is also a FiDiPro Professor (Finland Distinguished Professor Program (FiDiPro) supported by the Academy of Finland) at Tampere University of Technology, Department of Communications Engineering, Finland. He is the Editor-in-Chief of Computer Networks (Elsevier) Journal, and the founding Editor-in-Chief of the Ad Hoc Networks (Elsevier) Journal, the Physical Communication (Elsevier) Journal and the Nano Communication Networks (Elsevier) Journal. He is an IEEE Fellow (1996) and an ACM Fellow (1997). He received numerous awards from IEEE and ACM. His current research interests are in nanonetworks, 5G cellular systems, and wireless sensor networks.