IJSRD - International Journal for Scientific Research & Development| Vol. 2, Issue 05, 2014 | ISSN (online): 2321-0613

An Implementation of Minimum Energy Channel Codes For NanoScale Wireless Communications Using MAT Lab

Chethan Kumar B[1] Dr. Sarika Tale[2]
[1]Student [2]Associate Professor
[1,2]Department of DECS
[1,2]VTU Center for PG Studies, Bengaluru

Abstract— It is essential to develop energy-efficient communication techniques for nanoscale wireless communications. In this Proposal, a new modulation and a novel minimum energy coding scheme (MEC) are proposed to achieve energy efficiency in wireless nanosensor networks (WNSNs). Unlike existing studies, MEC maintains the desired code distance to provide reliability, while minimizing energy. It is analytically shown that, with MEC, codewords can be decoded perfectly for large code distances, if the source set cardinality is less than the inverse of the symbol error probability. Performance evaluations show that MEC outperforms popular codes such as Hamming, Reed- Solomon and Golay in the average codeword energy sense.

Key words: MEC, reliability , code distance , symbol error probability, WNSNs, Hamming code .Reed-Solomon, Golay codes.

I. INTRODUCTION

Wireless Nanosensor Networks (WNSNs), which are collections of nanosensors with communication capabilities, are believed to have revolutionary effects on our daily lives. A nanosensor is not necessarily a device merely reduced in size to a few nanometers, but a device that makes use of the unique properties of nanomaterials and nanoparticles to detect and measure new types of events in the nanoscale[1]. The development of novel communication techniques suitable for nanodevice characteristics is essential for WNSNs. One of the most promising building blocks for future nanodevices are carbon nanotubes (CNT). CNTs are rolled up graphene sheets with nano dimensions that can be used as nano-antennas, nano-sensing units and nano-batteries[2].

The resonant frequency of CNT antennas lies in the Terahertz band of the spectrum (0.1-10 THz). This band is not utilized by macro applications and is a candidate for communications between nanodevices. The main challenge of using the THz band is the absorption of EM waves by water vapour molecules, which makes communication impractical by causing severe path loss and molecular noise[4].

Employing channel coding at the nanoscale is critical to assure reliable communication between nanodevices.

The classical channel codes have various design considerations such as the efficient use of code space, as in perfect codes, bounded decoding complexity as the Shannon capacity is approached,as in Turbo or LDPC codes, or low encoding and decoding complexity as in cyclic and convolutional codes. However, the coding scheme for nano wireless communications should consider the *energy dissipation at the transmitter* as the main metric, Thus, classical codes are not suitable. Unlike most of the classical codes, minimum energy coding minimizes the average codeword energy, if OOK is the underlying modulation.

However, the existing minimum energy codes are unreliable. To address these needs, we develop a novel minimum energy channel code (MEC),that is reliable and suitable for nano communications. Proposed code provides the minimum average codeword energy of all the block codes, given that OOK is used as the modulation scheme.since nanonodes run on a strict energy budget. With OOK, average codeword energy is the symbol energy times average codeword weight; therefore, average energy is minimized by minimizing the average code weight. For this, codeword weights and sourceword-codeword mappings are chosen such that the expected code weight is minimized at the cost of increased codeword length, hence increased delay.

Lengthy codewords could increase the energy dissipation at the transmitter due to energy dissipation of the nanosensor circuitry. This implies a tradeoff between the transmission and processing energies and a discrete optimization problem could arise. However, such an analysis is not feasible today, since it is inaccurate to estimate the energy dissipation at the nano processing units, as no complete nanonode architecture is yet available. The suitability of MEC for nanoscale communications is shown by obtaining the achievable rate at the nanonode. Micro nodes act as central controller units of each cell to enable inter-cell communication and intra-cell Co-ordination. The maximum number of quantization levels and the effects of cell coverage ratio are investigated.

In nano-scale electronics have enabled the fabrication of nanoscale circuitries such as nano-transmitter, nano-receiver, and nano-processor.This envisages that current wireless technologies can be scaled down for plenty of new nano technology applications that is in Biomedical Field,Industrial Field,and Military Field. The hardware of a CNT sensor node includes four fundamental components, i.e., nano-transciever, nano-power, nano-processor, nano-memory, and nano-sensing units[4].

To address the low complexity requirement at the nanosensor nodes, lowcomplexity medium access techniques are investigated. Moreover, we develop four lemmas and the proofs of the Theorems[6]

This paper is organized as follows. In Section 2, we provide the basic structure Wireless NanoSensor Architecture .Section 3 describes Reliable medium Access technics. Section 4 shows the Related methodology of previous and existing method details. Section 5 gives the details of Channel Codes. Section 6 provides the details of M E C Codes .Simulation results and comparison with other reference implementations are discussed in section 7. Conclusions are summarized in the last part .

II. WIRELESS NANOSENSOR NETWORK ARCHITECTURE

Energy-efficiency and suitability for the THz channel are the prior concerns for the realization of WNSNs. Complexity of the nanosensor must also be kept as low as possible. The main functionalities of the nanonode structure shown in Fig. 2.1.

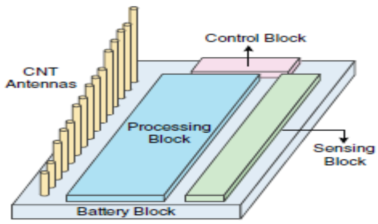

Fig. 1: Proposed nanosensor node architecture.

A multiple CNT antennas is proposed to utilize a number of available frequency windows in THz band. Required energy can be provided by the battery via nano energy-harvesting systems. Sensing is also CNT based. Nanosensor readings are quantized to M levels. No source coding is employed so as not to increase complexity. Each source signal level is mapped to $length - n$ channel codewords with a combinational nano-circuit. Realization of such a processing is not clear today. However, studies on CNT-based logic gate applications increase hope. The processing block is also responsible for carrier generation. Even though carrier generation in nano domain is not clear, it is shown that, with their unique properties such as slowing down surface EM waves, CNTs can also be used to generate THz waves much easier than the classical techniques. Control block contains a separate antenna for the control of the nanonode from a central unit. Nanonode activates and transmits only when this antenna is excited. This functionality is required for low complexity multiple access in WNSNs.

WNSNs can be used for sensing and data collection with extremely high resolution and low power consumption in various applications.

A. *Multi-carrier OOK Modulation:*

Each codeword is transmitted in parallel over different carriers. Our frequency choice considers carriers' suitability for transmission in the THz channel. The THz channel consists of several frequency windows with low absorption and low molecular noise, termed as *available windows*, which depends on the transmission distance and water vapour amount on the transmission path. Carrier frequencies are chosen among these windows in the THz channel. CNTs are used as nano-antennas to radiate each carrier, as shown in Fig. 2.1. Each frequency window is utilized separately. Bandwidth increase is prohibited by the molecular absorption lines. Decreasing the bandwidth results in increased energy consumption per symbol, since symbol duration increases.

Hence, we select bandwidth as the same as the width of the available frequency windows. Hence, picoseconds long sinusoidal pulses are used, which span a frequency band of 100-200 GHz, corresponding to the width of most of the windows in the THz channel.

B. *B. WNSN Cell Architecture:*

A cell-based WNSN is considered,a cell is composed of micro node, and nanosensor nodes scattered around it. In order to reduce the interference, nanonodes are deployed within a radius of αR, where R is the cell radius and α is called the coverage ratio satisfying $0 < \alpha \leq 1$. To keep the complexity of the nanonodes low, all the control and scheduling issues are left to the micro node within the cells. A nanonode starts transmission only when an activation signal is sent by the micro node. KHz band can be used for this activation signal, with vibrating CNTs. The central micro node provides not only control, but also synchronization among the nanosensors. It is assumed that the micro node is capable of receiving the THz waves. CNT bundles can be used for efficient THz detection at room temperature.

III. RELIABLE MEDIUM ACCESS TECHNIC

A. *Single Control Signal:*

Nanonodes start transmission simultaneously through disjoint sets of channels (frequencies). To keep complexity at the micro node, different sets of frequencies must be used by each nanonode, and a common synchronization signal must be broadcast from the micro node for signalling the transmission. Nl different THz frequency windows, and a single kHz band are allocated to a single cell. This is an FDMA-based scheme, as separate frequency windows are allocated to each nanosensor node.

B. *Multiple Control Signals:*

Nanonodes use the same set of frequencies for transmission. The micro node uses control signals at different frequencies for each nanonode sequentially, as nanonodes utilize the same THz channels. Allocation of l THz and N kHz bands are needed. This is similar to TDMA, since all the nodes use the channel in different time intervals. In the following, we assume that the micro node uses multiple control signals, since the number of frequency windows in the THz channel is limited and demodulating a number of different THz signals significantly increases complexity.

IV. RELATED METHODS

A. EXISTING METHOD:

Employing channel coding at the nanoscale is critical to assure reliable communication between nanodevices. The classical channel codes have various design considerations such as the efficient use of code space, as in perfect codes, bounded decoding complexity as the Shannon capacity is approached, as in Turbo or LDPC codes, or low encoding and decoding complexity as in cyclic and convolutional codes. However, the coding scheme for nano wireless communications should consider the energy dissipation at the transmitter as the main metric, since nanonodes run on a strict energy budget. Thus, classical codes are not suitable. Unlike most of the classical codes, minimum energy coding minimizes the average codeword energy, if OOK is the underlying modulation. However, the existing minimum energy codes are unreliable.

B. PROPOSED METHOD:

A novel minimum energy channel code (MEC), that is reliable and suitable for nano communications is develpoed. Proposed code provides the minimum average codeword energy of all the block codes, given that OOK is used as the modulation scheme. With OOK, average codeword energy is the symbol energy times average codeword weight; therefore, average energy is minimized by minimizing the average code weight. For this, codeword weights and sourceword-codeword mappings are chosen such that the expected code weight is minimized at the cost of increased codeword length, hence increased delay. Lengthy codewords could increase the energy dissipation at the transmitter due to energy dissipation of the nanosensor circuitry.

V. CHANNEL CODES

A. Repetition:

The straight forward solution to this problem is to repeat each bit a certain n times and then use the majority as the correct bit. For example, with n = 3 the message 101 would be sent as 111000111. In order for this to be useful, the message must be repeated at least three times. It is easy to see that if less then n2 errors occur in each group then the message will be decoded correctly. As the number of repetitions increases, the probability of an undecipherable error decreases proportionately. Unfortunately, this also correlates with an increase in the amount of data being sent. This is a very crude method of error correction. It works most actively when there is unlimited bandwidth and it is essential that the data is recoverable. But it is unacceptably costly on a practical scale. Also if a user needs to encrypt the data for security, knowing that the information is repeated allows for significantly easier cracks. Therefore more sophisticated repetition algorithms are needed.

B. Parity Checkers:

Rather then repeating every bit, it would be more efficient if we encoded and verified chunks of data. A solution to that is to introduce a parity-check bit. Parity check bits are bits that demonstrate whether another set of bits is even or odd. By tradition the parity-check bits attempt to make the number of 1's even. For example, if we were attempting to send the string 001 then adding a parity-check bit to the end to check the entire string would cause the new string to become 0011. Therefore if one bit is flipped, a parity-check bit would be able to tell. However, it would not be able to determine what bit was flipped or if two bits were flipped. Therefore a single parity-check bit could only inadequately check the accuracy of a message and would be unable to fix it. More sophisticated methods layer multiple parity bits to gain corrective capabilities.

C. Hamming Codes:

What is generally consider the first leap towards error correcting codes was done by R. W. Hamming during his employment at Bell Labs. He wrote his paper in 1950, but improved Hamming Codes are still in use today. He began by detailing the use of a single parity-check bit, but quickly determined that more sophisticated methods would be needed in order to correct errors.

Hamming Codes send *m* information bits padded with a specific *k* parity-check bits. They have the ability to correct any single mistake. They manage this by having the *k* parity-check bits set at positions $1,2,\ldots,2^{k-1}$ and checking every element whose binary representation has a "1" in position *ki*-1. For example, bit 4 would check the sum of the parities in positions 100, 101, 110, 111, 1100, 1101…. = 4, 5, 6, 7, 12, 13,….

D. Reed-Muller and Reed-Solomon Codes:

Reed-Muller and Reed-Solomon Codes use vectors that partially span a vector space as their way of inducing repetition. They create a polynomial using the data bits as the coefficients for the spanning vectors. Then they send an oversampled section of this polynomial. The original polynomial can be reconstructed by multiplying the data points with vectors perpendicular to the spanning vectors. Then, the original data can be reconstructed. This description is slightly misleading. Both Reed-Muller and Reed-Solomon codes use this idea as the basis, but their implementations vary widely. In particular, Reed-Muller codes are designed to only handle binary representation and they approach the matter differently.

It has the very convenient fact that the larger the code word, the more errors this code can correct. It also has the ability to tailor exactly how many correctable errors at the cost of more sent data. Without going into too much detail, Reed-Muller codes are described as $R(r;m)$, where m is the number of spanning vectors (causing the space to have $2m$ dimensions) and r is the depth of linear combinations of spanning vectors. For example, $R(2;4)$ has 16 dimensions, and is partially spanned by vectors $x1$, $x2$, $x3$, $x4$ and $x1x2$, $x1x3$, $x1x4$, $x2x3$, $x2x4$, $x3x4$. It can therefore encode a message of up to 10 bits long and the sent message would be 16 bits long. The minimum distance between codewords is 2^{m-r}. So the exact number of errors any Reed-Mullen code can correct is $\frac{2^{m-r}-1}{2}$ 2 rounded down. In the above example the 10 bit code could have corrected $\frac{2^{4-2}-1}{2} = 1$ error.

E. E. Low-Density Parity-Check Codes (LDPC):

Low-density parity-check (LDPC) codes are encodings that use specific parity bits. They are designed in such a way that all bits act equivalently. Each parity-check bit checks some small fixed $k \in Z$ bits and each bit is checked by some small fixed $j \in Z$ parity-check bits. LDPC codes are also highly efficient.

The major contribution of LDPC codes at the time they were invented was their decoding algorithm. Every prior decoding method was strict and deterministic. Every potential outcome had an exact procedure to follow that lead to a single solution. LDPC implements two different "softer" decoding algorithms. The first decoding algorithm cycles through the digits and checks each one versus its parity check operators. If the majority of them contradict what the bit is, then it is flipped. This process is repeated until all bits are static. The second approach is more accurate and computationally intensive. It proceeds similarly to the first approach, but instead of using a majority process it computes the probability that a certain bit is a 1, by taking into account all the other bits. Then it iterates through until it reaches a static position. In both these, the results depend upon the order the elements are examined. These decoding algorithms were computationally impractical at the time they were created.

F. Convolution Codes:

Up until now, each individual block has been encoded as an independent entity. Convolution codes encode bits based upon a state which is determined by summing a fixed set of previously bits. Each input bit is manipulated in a few different ways to produce several outputs bits. Therefore each output bit conveys the combined information of many different input bits. The state is initialized to a key that is initially passed from encoder to decoder. Due to the integral part this key plays in decoding, these codes are often used for cryptography.

The codes can be further subdivided into systematic and recursive: a systematic code has an output that is the input; a recursive algorithm uses a prior output as part of the new input. Figure(a) depicts three diagrams of convolution encodings. They all use a rate of 1/2 (or 1 input, 2 outputs) and have a state that is dependent upon seven inputs. The top image depicts a systematic nonrecursive, the middle a nonsystematic nonrecursive, and the bottom a systematic recursive. Since there are so many different types of convolution codes it is diffcult to do direct comparisons. But overall, convolution codes are significantly better at approaching the theoretical Shannon limit then prior error correcting codes. They are fast, effcient and generally accurate. Unfortunately their accuracy varies significantly depending on the input. In specific, convolution codes have specific codewords where their accuracy plummets. Some codewords are only separated by a distance of one. Much research in this field attempts to reduce the quantity of these trouble words. However, creating a convolution that is free of them has yet to be done.

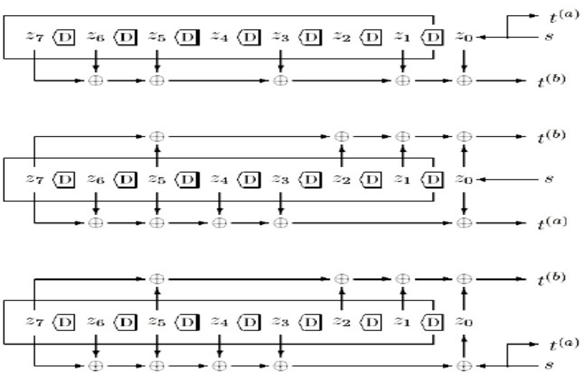

Fig. 2: Different methods of generating convolutional codes with rate 1/2 .

The symbol D indicates a copying with a delay of one clock cycle. The symbol ⊕ denotes linear addition modulo 2 with no delay. The symbol *s* denotes the input. The symbol *t*(*b*) and *t*(*a*) represent the two outputs.

Convolution codes allow for the addition of other checksums. It is common practice to integrate a form of Reed-Muller parity-checks into the code before convoluting it. This gives double error-protection, but sacrifices some of the speed associated with encoding and decoding convolution codes.

V. MINIMUM ENERGY CHANNEL CODING

This work proposes a new channel code, which minimize the average code weight. Such codes are equivalent to the codes minimizing average codeword energy for the systems employing OOK modulation. This is because, no energy is dissipated when 0 symbol is transmitted and no ARQ scheme is employed in nano communications for retransmissions. For block codes, a codebook is defined as any selection of fixed length codewords, mapped to source symbols.

For unique decodability, this mapping should be one-to-one. Weight is the number of non-zero entries in the codeword. As we deal with binary codes, weight is equivalent to the number of 1s in the codeword. Weight enumerator of a code is the polynomial $W_C(z) = \sum_i c_i z^i$ where c_i is the number of codewords with weight i.

Additionally, the distance (or Hamming distance) between two codewords is defined as the number of bits in which they differ.. In minimum distance decoding, which is the presumed decoding strategy, the received n-tuple is mapped to the closest codeword. Codes with distance d can correct $\lfloor d-1/2 \rfloor$ errors, and reliability increases with distance, since more error patterns can be corrected. Codewords with lower weight results in less energy dissipation, when transmission of 0 symbol requires less energy than the transmission of 1 symbol In the nanonodes, each codeword has the same probability of occurrence as the source outcomes that they are mapped to, since no source coding mechanism is employed.

Let M, d, p_{max} represent number of codewords, code distance, maximum probability in any discrete distribution and the source random variable, respectively.

A. Lemma 1:

For any finite M, there exists a finite n_0 such that a constant weight code C of length-n0 containing the codeword c can be constructed with code distance d, if and only if $weight(c) \geq \lceil d/2 \rceil$:

$$\exists \mathbb{C} : dist(\mathbb{C}) \geq d \text{ for } c \in \mathbb{C} \Leftrightarrow weight(c) \geq \lceil d/2 \rceil$$

B. Lemma 2:

Any codebook with code distance of d contains at most a single codeword with weight less than $\lceil d/2 \rceil$

C. Lemma 3:

Any two codeword c_i and c_j of a code with distance d should satisfy the inequality $weight(c_i) + weight(c_j) \geq d$.

Let C_i be the code with weight enumerator

$$W_{\mathbb{C}_i}(z) = z^{\lfloor \frac{d}{2} \rfloor - i} + (M-1)z^{\lceil \frac{d}{2} \rceil + i} \quad \ldots\ldots (1)$$

The code C_i contains a single codeword with weight $\left\lfloor \frac{d}{2} \right\rfloor - i$ and all the other codewords have weight $\left\lceil \frac{d}{2} \right\rceil + i$. Let codeword with weight $\left\lfloor \frac{d}{2} \right\rfloor - i$ be assigned to the source symbol with maximum probability, i.e., p_{max} . Let E_{C_i} represent expected code weight for code C_i.

$$E_{C_{i+k}} < E_{C_i} \text{ if } p_{max} > 0.5, \forall k > 0.$$

D. Lemma 4:

Proof: Let β represent $\left\lfloor \frac{d}{2} \right\rfloor$. Then

$$E_{C_i} = p_{max}(\beta - i) + (1 - p_{max})(d - \beta + i)$$
$$= p_{max}(2\beta - 2i - d) + d - \beta + i.$$
$$\Rightarrow E_{C_i} - E_{C_{i+k}} = k(2p_{max} - 1).$$

Hence, since k is positive, $E_{C_i} + k < E_{C_i}$ *if* $p_{max} > 0.5$.

E. Theorem 1:

Let $X = x_i$ has probability $p_i \in \{p_1, p_2, \dots, p_M\}$ and p_{max} be $max(pi)$. For a desired code distance d, the minimum expected codeword weight, $E(w)$ is

$$\min(E(w)) = \begin{cases} (1 - p_{max})d, & p_{max} > \frac{1}{2}, \\ \frac{d}{2}, & p_{max} < \frac{1}{2}, \ d \ even, \\ \left\lceil \frac{d}{2} \right\rceil - p_{max}, & p_{max} < \frac{1}{2}, \ d \ odd \end{cases} \quad \text{......(2)}$$

Proof: Let c_i be the codeword assigned to the source symbol x_i that has probability p_i and w_i represent $weight(c_i)$. From Lemma 1, a $weight - \left\lceil \frac{d}{2} \right\rceil$ code can be constructed with finite code length for any M. Therefore, $n(E(w)) \leq \left\lceil \frac{d}{2} \right\rceil$. From Lemma 2, we know that we can decrease the weight of only a single codeword below $\left\lceil \frac{d}{2} \right\rceil$. Then the bound can safely be improved by switching the code weight of the most probable outcome to $\left\lfloor \frac{d}{2} \right\rfloor$, since the resultant code will still satisfy the distance condition. This leads to a bound valid for any source probability distribution:

$$\min(E(w)) \leq p_{max} \lfloor d/2 \rfloor + (1 - p_{max})(d - \lfloor d/2 \rfloor) \quad \text{....(3)}$$

From Lemma 3, to further reduce the weight of the most probable codeword, we should increase the weight of all the other codewords to satisfy $weight(c_i) + weight(c_j) = d$ for any i, j. Lemma 4 shows that this operation, i.e., increasing i in (1), decreases the average weight, if $p_{max} > 0.5$. Hence, minimum average weight is obtained when $i = \left\lfloor \frac{d}{2} \right\rfloor$ for $p_{max} > 0.5$. This yields the code C_d with the enumerator $W_{C_d}(z) = z^0 + (M - 1)z^d$, giving the average weight

$$E(w) = (1 - p_{max})d. \quad \text{.......(4)}$$

Note that by mapping the codeword of $\left\lfloor \frac{d}{2} \right\rfloor$ weight to any symbol with probability $p < p_{max}$, this bound cannot be reached, since decreasing the weight of this chosen codeword does not decrease the average weight as $p < p_{max}$ forces $p < 0.5$.

F. Theorem 2:

Let $X = x_i$ has probability $p_i \in \{p_1, p_2, \dots, p_M\}$ and p_{max} be $max(pi)$.. For a desired code distance d and maximum codeword weight k, if $\left\lceil \frac{d}{2} \right\rceil \leq k < d$ is satisfied, minimum expected codeword weight, $E(w)$ is given by

$$\min(E(w)) = \begin{cases} p_{max}(d - 2k) + k, & p_{max} > \frac{1}{2}, \\ \frac{d}{2}, & p_{max} < \frac{1}{2}, \ d \ even, \\ \left\lceil \frac{d}{2} \right\rceil - p_{max}, & p_{max} < \frac{1}{2}, \ d \ odd \end{cases} \quad \text{...(5)}$$

Proof: It is clear that, if $p_{max} < 0.5$, bound given in Theorem 1 can be achieved, since $k \geq d/2$. However, if $p_{max} > 0.5$, by Lemma 4, i in (1) should be increased to reduce the average code weight, and could at most be $i = k - \left\lceil \frac{d}{2} \right\rceil$ due to weight constraint. Hence, for $\left\lceil \frac{d}{2} \right\rceil \leq k < d$,

$$\min(E(w)) = p_{max}(d - k) + (1 - p_{max})k. \quad \text{....(6)}$$

Combining both cases, theorem is obtained in few steps

G. MEC PARAMETERS:

1) Power Dissipation:

Power dissipated for codeword i is $P_i = w_i P_{sym}$, where P_{sym} is the symbol power. Then the average power is

$$E(P) = \sum_{i=1}^{M} w_i p_i P_{sym} = E(w) P_{sym}. \quad \text{....(6)}$$

Equation (6) also shows the average power per log(M) bits, since codewords carry log(M) bits of information. For different source distributions, information per codeword will be different from an information theoretic point of view. However, for simplicity, we assume each codeword carries log(M) bits of information, leaving the information theoretic analysis to a future study.

2) Minimum Codeword Lengt:

n_{min} is the minimum codeword length required to satisfy the MEC weight enumerator for given M and d. n_{min} is important as it yields the minimum delay due to transmission of codewords. $A(n, d, w)$ is the maximum number of codewords of length n with code distance d and fixed code weight w.

$p_{max} < 0.5, d\ even$: Weight enumerator of MEC is $W_C(z) = Mz^{\frac{d}{2}}$. Therefore, $n_{min} = min\{n : A(n, d, d/2) \geq M\}$. Since 1s in each codeword are disjoint, $n_{min} = \frac{Md}{2}$.

$p_{max} < 0.5$, $d\ odd$: From Theorem 1, we know that the weight enumerator is $W_C(z) = z^{\left\lfloor \frac{d}{2} \right\rfloor} + (M - 1)z^{\left\lceil \frac{d}{2} \right\rceil}$. 1s in all the codewords should be disjoint with the 1s in the most probable codeword, i.e., the codeword with weight $\left\lfloor \frac{d}{2} \right\rfloor$. Hence, $n_{min} = \left\lfloor \frac{d}{2} \right\rfloor + min\{\tilde{n} : A(\tilde{n}, 2m + 1, m + 1) \geq M - 1\}$, where $d = 2m + 1$. The following property is helpful:

$$A(n, 2m - 1, w) = A(n, 2m, w)$$
$$\Rightarrow A(\tilde{n}, 2m + 1, m + 1) = A(\tilde{n}, 2m + 2, m + 1). \quad \text{.....(7)}$$

Therefore, $min\{\tilde{n}\} = (m+1)(M-1)$.
Hence, $n_{min} = m + (m+1)(M-1) = \left\lceil \frac{d}{2} \right\rceil M - 1$.

– $p_{max} > 0.5$: In this case, MEC has the weight enumerator $W_C(z) = z^0 + (M-1)z^d$ and maps the all-zero codeword to the most probable source event. Minimum codeword length is found as $n_{min} = min\{n : A(n,d,d) \geq M-1\}$. In the literature, there is no explicit formulation for $A(n,d,d)$. We can use the existing lower bounds on the code size.

$$A(n,2m,w) = A(n,2m-1,w) \geq \frac{1}{q^{m-1}}\binom{n}{w}$$
$$\Rightarrow A(n,d,d) \geq \frac{1}{q^{\lceil \frac{d}{2} \rceil - 1}}\binom{n}{d}, \quad \ldots(8)$$

where q is a prime power such that $q \geq n$.

The codewords for $p_{max} < 0.5$ and $d-$ even case can be constructed by cyclic shifting of a $d/2 - length$ block of 1s by an amount of $d/2$. Based on this cyclic shifting idea, we have developed a code construction scheme. In this approach, blocks of 1s are shifted by proper amounts to satisfy the Hamming distance with the previous codeword. The obtained minimum codeword length under such a construction is

$$n_{min} = d + (M-2)\lceil d/2 \rceil \quad \ldots..(9)$$

H. Error Resilience:

The received n-tuples are mapped to the codeword to which they are closest in terms of Hamming distance. Then the probability that codeword is correctly decoded is

$$\xi_d = \sum_{i=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{n_{min}}{i} p_s^i (1-p_s)^{n_{min}-i}. \quad \ldots(10)$$

It is been shown that for sufficiently large distance, codewords are correctly decoded with high probability, if the symbol error probability is less than the inverse of source set cardinality.

$$\xi = \lim_{d \to \infty} \xi_d = \begin{cases} 1, & p_s < 1/M \\ 0, & p_s > 1/M \end{cases}. \quad \ldots(11)$$

Equation (11) is proven in Appendix. Hence, perfect communication can be achieved among nanosensor nodes and micro node, if $M < 1/p_s$, by keeping the code distance sufficiently large. Hence, if symbol error probability is decreased, nanosensor readings can be quantized with smaller quantization steps.

The micro node utilizes coherent detection and hard decoding to detect the transmitted symbol. Therefore, symbol error probability is given as $p_s = 0.5[1 - erf\left(\left(\frac{A2}{8\sigma 2n}\right)^{0.5}\right)]$,

$$P_r = \frac{P_{sym}}{A(f,r)} = \frac{A^2}{2}, \ \sigma_n^2 = k_B TB + P_{sym} \sum_{i \in S} \frac{1}{A(f_i, r_i)} \quad \ldots.(12)$$

where kB, T, B, r are Boltzmann constant, temperature, bandwidth and transmission distance. $A(f,r) = \left(\frac{4\pi f r}{c}\right)^2$ is the loss term, where f is frequency and c is the speed of light.

I. Energy per Information Bit:

Next, we obtain energy per information bit to demonstrate the energy efficiency of our coding scheme. Probability that a codeword is correctly decoded, which is obtained in (10), can also be obtained as follows using law of large numbers:

$$\xi_d \approx \frac{\#\text{ of codewords correctly decoded}}{\#\text{ of codewords transmitted}} \quad \ldots.(13)$$

for a large number of transmitted codewords, for a code with distance d. Therefore, the average energy per bit is expressed as the ratio $\frac{EcQ}{log(M)Q\xi_d}$, i.e.,

$$\eta = \frac{E(w)P_{sym}T_{sym}}{\log(M)\xi_d} \ joules/bit. \quad \ldots(14)$$

VI. SIMULATION RESULTS AND DISCUSSIONS

A. Minimum code weight vs. source mean:

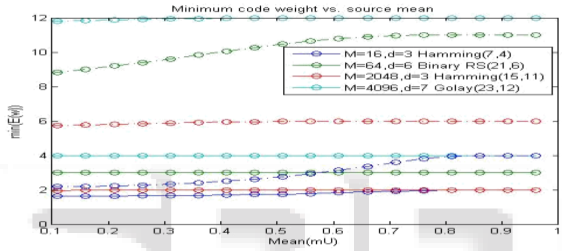

Fig. 3: Minimum code weight vs. source mean

MEC is compared with the classical block codes in Fig.(i) in terms of expected code weight. To minimize code weight for the Hamming, Reed-Solomon and Golay codes, more probable source symbols are assigned to codewords with less weight, using the corresponding weight enumerators. We use the binary expansion of 8-ary (7,2) Reed Solomon code.

B. Correct codeword decoding vs. symbol error:

Codeword decoding performances of MEC, Golay and Hamming codes are illustrated in Fig. ii(a)- ii(c). MEC is not as effective as the others in terms of error correction. This is due to the different codeword lengths. Lengthy codes have more uncorrectable error patterns, which decreases the error correction probability. As observed in Fig.ii, correct decoding probability increases with code distance and approaches to 1, if symbol error probability, p_s, is less than the inverse of source set cardinality, $1/M$.Intuitively, transmitted information increases with M, which requires more reliable channels.

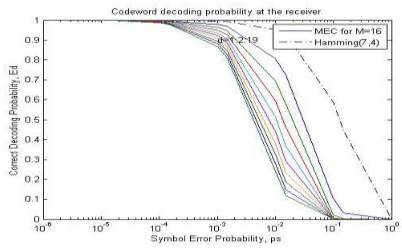

Fig. 4: Codeword decoding probability at the receiver with M=16 and 1<d<19

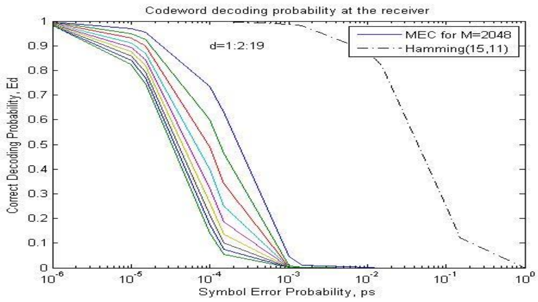

Fig. 5: Codeword decoding probability at the receiver with M=2048 and 1<d<19

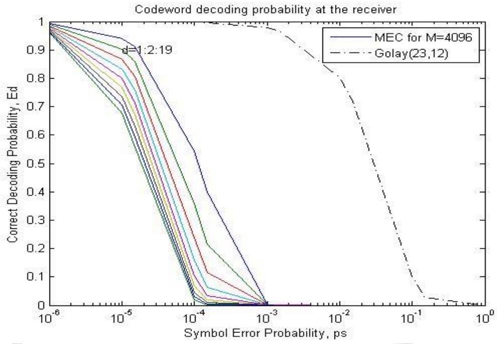

Fig. 6: Codeword decoding probability at the receiver with M=4096 and 1<d<19

C. Energy efficiency vs. symbol error:

The average energy per received bit, i.e., η as is shown in Fig. iii (a)-iii (c) for a symbol energy of $10^{-5}\ pJ$. Samples of a Gaussian distribution with $\sigma = 0.5$ are taken and normalized. η is calculated for each case separately. MEC is better in terms of average energy per bit for symbol error probabilities less than a threshold. As p_s exceeds the threshold, average energy per bit exponentially increases, since correct codeword decoding is unlikely. Note that the observed behavior is dominated by $1/\xi$ factor .

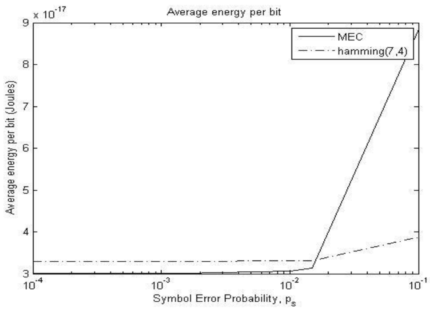

Fig. 7: Average energy per bit with M=16

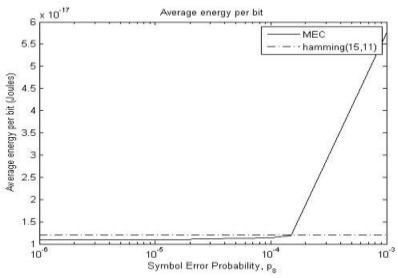

Fig. 8: Average energy per bit with M=2048

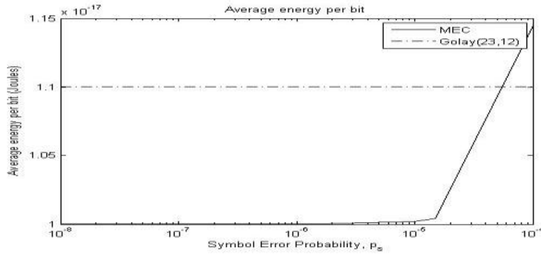

Fig. 9: Average energy per bit with M=4096

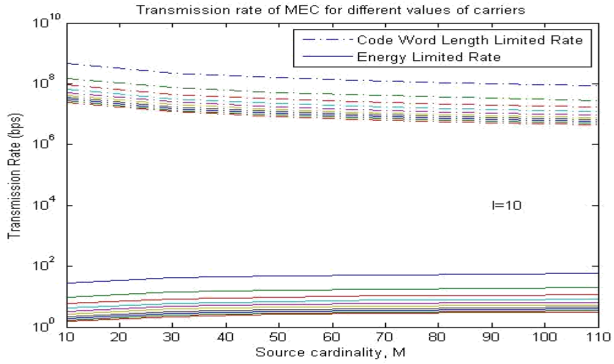

Fig. 10: Transmission rate of MEC for different number of subcarriers (l=1)

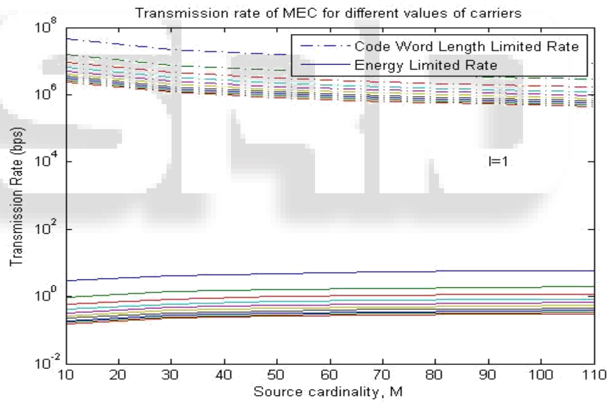

Fig. 11: Transmission rate of MEC for different number of subcarriers (l=10)

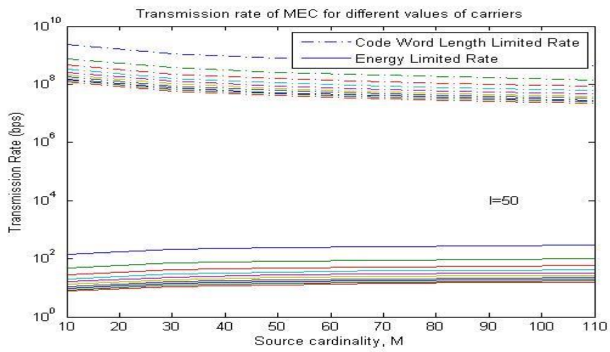

Fig. 12: Transmission rate of MEC for different number of subcarriers (l=50)

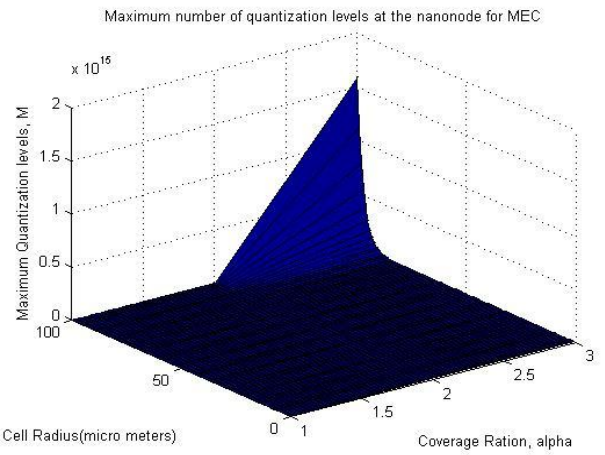

Fig. 13: Maximum number of quantization levels at the nano-node for MEC

VII. CONCLUSIONS

In this work a multi-carrier OOK modulation was proposed, motivated with the THz channel characteristics, and develop a novel minimum energy channel code, MEC, for nano communications in cell-based WNSNs. MEC satisfies a minimum Hamming distance to guarantee reliability. It is analytically shown that codewords can be decoded perfectly using MEC with large code distance, if the number of quantization levels is less than the inverse of symbol error probability. Simulations show that, the proposed code is superior to popular block codes such as Hamming, Reed-Solomon and Golay. The stateof- the-art nanoscale power and energy limits are used to obtain achievable rates of nanonodes, which are expected to be on the order of Mbps, neglecting the processing power. Numerical results show that MEC is an energy-efficient and reliable code for future WNSNs with cell radius up to several millimeters.

REFERENCES

[1] I. F. Akyildiz and J. M. Jornet, "Electromagnetic wireless nanosensor networks," Nano Commun. Net., vol. 1, pp. 3–19, 2010.
[2] G. W. Hanson, "Fundamental transmitting properties of carbon nanotube antennas," IEEE Trans. Antennas Propag., vol. 53, pp. 3426–3435, 2005.
[3] B. Atakan and O. B. Akan, "Carbon nanotube sensor networks," in 2009 IEEE NanoCom.
[4] J. M. Jornet and I. F. Akyildiz, "Channel capacity of electromagnetic nanonetworks in the Terahertz band," in 2010 IEEE ICC.
[5] C. Erin and H. H. Asada, "Energy optimal codes for wireless communications," in 1999 IEEE CDC.
[6] M. Kocaoglu and O. B. Akan, "Minimum energy coding for wireless nanosensor networks," in 2012 IEEE INFOCOM.
[7] M. Kocaoglu and D. Malak, "On the node density limits and rate-delay energy tradeoffs in ad hoc nanonetworks with minimum energy coding," in 2012 IEEE MoNaCom.
[8] J. M. Jornet and I. F. Akyildiz, "Low-weight channel coding for interference mitigation in electromagnetic nanonetworks in the terahertz band," in 2012 IEEE ICC.
[9] Y. Prakash and S. K. S. Gupta, "Energy efficient source coding and modulation for wireless applications," in 2003 IEEE WCNC.
[10] S. Xu et al., "Self-powered nanowire devices," Nat. Nanotechnology., vol. 5, pp. 366–373, 2010.
[11] Y. Huang et al., "Logic gates and computation from assembled nanowire building blocks," Science, vol. 294, pp. 1313–1317, 2001.
[12] O. V. Kibis et al., "Generation of terahertz radiation by hot electrons in carbon nanotubes," Nano Lett., vol. 7, pp. 3414–3417, 2007.
[13] I. F. Akyildiz and J. M. Jornet, "The Internet of nano-things," IEEE Wireless Commun., vol. 17, pp. 58–63, 2010.
[14] K. S. Yngvesson et al., "Experimental detection of terahertz radiation in bundles of single wall carbon nanotubes," in 2008 ISSTT.
[15] F. J. MacWilliams and N. J. A. Sloane, The Theory of Error-Correcting Codes. North-Holland, 1977.

