Efficient Power Saving Algorithm to Detect Collision Based on Statistics of Received Packets in Wireless Sensor Networks

Fawaz Alassery, Student, IEEE, Walid K. M. Ahmed, Senior, IEEE, Mohsen Sarraf, Senior, IEEE and Victor Lawrence, Fellow, IEEE

Abstract— **Recently a lot of research effort has been focused on Wireless Sensor Networks (WSNs) due to its various applications. Over the last few years, several techniques have been proposed for investigating the power consumption which represents one of the most challenges and main concerns in designing WSNs. Power consumption of nodes in WSNs has a great effect on the lifetime of network nodes which are difficult to replace or recharge. In this context, this paper represents a receiver approach for alleviating power consumption of WSNs. Unlike other power consumption techniques, instead of decoding every received signal at the receiver which consume too much power our approach studies the histograms of sensors' transmitted signals to detect collisions, so the receiver can determine when the transmitted signals can be decoded without wasting precious power decoding transmitted signals suffering from collisions .Based on a set of algorithm metrics, thresholds and scenarios, our approach shows reduction in power consumption. We use MATLAB to show our power consumption performance gains.**

Index Terms— **WSN, Power Consumption Techniques, WSN Protocols, Packets Collision**

I. INTRODUCTION

WIRELESS Sensor Networks (WSNs) consist of many sensor nodes distributed in various environments in order to perform specific tasks such as passive localization, target tracking, systems control, healthcare monitoring, air pollution and temperature monitoring, irrigation management and water monitoring, etc.

In many cases each node in a WSN has a limited power source which is a small battery. After the initial deployment of nodes in an environment, the nodes must be active for a long period of time. Therefore, power saving techniques play a very important role in order to extend the lifetime of WSN nodes [4].

There are many aspects that lead to waste of energy in WSNs. These aspects affect the efficacy and efficiency of WSNs. One example is when a collision occurs between two or more transmitted packets and the receiving node decodes the received signal to detect the collision. In addition, the interference, overhearing and unnecessary retransmission of packets from different nodes consumes too much power [1].

In term of power consumption in WSN transmission, it is obvious that the transmissions in a WSN follow different stages either in a transmitter or a receiver. Each stage has its own electronic circuit which consumes some power. Power consumption starts from baseband processing of data, amplification, filtering, modulation and RF front-end stages of the transmitted signal in a transmitter to RF front-end and demodulation processes in the receiver. In general, the sensor node has hardware blocks which cause energy loss due to the specific function that is performed by that block [5].

The remainder of this paper is organized as follows. Section II investigates related work. Section III describes our proposed system. In Section IV we define the algorithm and metrics, and show how to select the system thresholds. In section V, we compare the computational complexity of our metrics against commonly used decoding technique (Soft Output Viterbi Algorithm-SOVA). In section VI we show and discuss the results of the simulations used to model our system for different scenarios. Finally, we offer our conclusions in section VII.

II. RELATED WORKS

Many techniques have been introduced in various studies aimed at maximizing WSN node lifetime by reducing power consumption. Variety of definitions for WSNs lifetime is introduced based on network connectivity, coverage, application requirements, and number of active nodes [2] [3]. Power efficient techniques in WSNs have been categorized into five classes briefly introduced in the following:

First class is the power efficient techniques that focus on reducing the data processed and transmitted from the source sensor. In [6], authors use clusters in order to aggregate the information being transferred. They proposed LEACH (Low-Energy Adaptive Clustering Hierarchy) which is a clustering based protocol aimed to distribute the energy load among the WSNs nodes. Another data reduction strategy proposed in [7] which is based on avoiding transfer of the information to undesirable sensors, this can be done via defining a smaller dominating set of sensors when two hops are considered. Moreover, data reduction can be resulted from compression algorithms explained thoroughly in [8] where authors investigate compression algorithms applicable in WSNs such as Coding by Ordering, Pipelined

Manuscript received March 14, 2014; revised March 28, 2014.

Fawaz Alassery is with the Electrical and Computer Engineering Department (ECE), Stevens Institute of Technology. Hoboken, NJ 07030 USA. Phone: 201-912-7162; e-mail: falasser@stevens.edu.
Walid K. M. Ahmed is with ECE Department, Stevens Institute of Technology. Hoboken, NJ 07030 USA. e-mail: walidmail@yahoo.com.
Mohsen Sarraf is with ECE Department, Stevens Institute of Technology. Hoboken, NJ 07030 USA. e-mail: ms7463@yahoo.com
Victor Lawrence is with ECE Department, Stevens Institute of Technology. Hoboken, NJ 07030 USA. e-mail:victor.lawrence@steven.edu

In-Network Compression, Low-Complexity Video Compression and Distributed Compression.

Second class of power efficient techniques in WSNs deals with controlling the topology via tuning the transmission power while maintaining the connectivity of the network. In this context, authors in [9] present a Local Minimum Spanning Tree (LMST) algorithm to control the wireless multi-hop topologies. In the mentioned algorithm each node builds its own LMST independently using locally collected data. This algorithm leads to further increase in network capacity and a significant reduction in power consumption. Furthermore, an Adaptive Transmission Power Control (ATPC) algorithm for WSNs is proposed in [10]. Each node in the network builds a model which describes the correlation between the transmission power and link quality with its neighboring nodes.

Reducing unsuccessful end-to-end transmissions and avoiding nodes that consume too much power in routing packets of WSNs is the third class of power efficient techniques. Some protocols in this class use the advantages of mobility and broadcast communication to reduce the power consumption when sending packets to a sink node. Others protocols use the geographical coordination of source nodes to determine their position when building the route that connects them to destination nodes [2]. In [11], authors proposed energy aware routing algorithm that take into account the interference that may occur from neighboring nodes in multi-hop wireless networks. The algorithm automatically routes around the congested areas which has a significant impact on controlling congestion in the network. In [12] authors propose Direct Diffusion routing protocol that constructs a new routing tree via Geocast approach when failures occur in the routing path. The simulation of their proposed protocol shows a reasonable reduction in power consumption. A survey on energy aware routing protocols in WSN is provided in [13].

The fourth class of power efficient techniques schedules the sleeping states of sensor nodes and alternates to active states while maintaining the network application functionalities. One method in this class is explained in [14] where sensor nodes are organized in set covers. Monitoring targets is performed at specific time by sensors in only one set while the sensors in every other set are in sleeping mode. Authors in [15] use TDMA as MAC protocol to propose an algorithm called contiguous link scheduling (assigning one time slot to each sensor). This algorithm takes advantage of avoiding collisions using TDMA slots to reduce the frequency of state transitions.

Energy efficient modulation schemes and ingenious coding processes can be considered as the fifth class of power efficient techniques in WSNs. In fact, optimal selection of modulation schemes and intelligent coding techniques improve both the energy and bandwidth efficiency. For example, Adaptive Modulation and Coding (AMC) explained in [16] yield higher performance over long distances. Moreover, to test Bit Error Rate (BER) and power consumption in WSNs, many error control coding techniques such as RS and BCH codes are discussed in [17]. In [18], authors show that the binary BCH code with ASIC implementation in WSNs outperform other types of codes such as RS and convolutional codes where they use random data run through a Gaussian channel to compare the BER performance of three error control codes. In addition, with respect to BER, energy and lifetime performance parameters over shorter distances with AWGN and Rayleigh fading channels, a comparison between some modulation schemes such as MPSK, M-QAM and M-FSK is detailed in [19]. Authors reach the conclusion that M-QAM is more efficient for WSNs. Reference [20]; further compares various error control codes resulting in reasonable power saving at the transmitter at the cost of increasing power consumption in decoding in the receiver.

III. SYSTEM DESCRIPTION

Our system contains a variable number of sensors that can be deployed anywhere to perform their functions (e.g., sensing, monitoring, etc.). These sensors send their collected data to a central sensor for further processing or communicating with other networks. Figure1shows a high level view of this network.

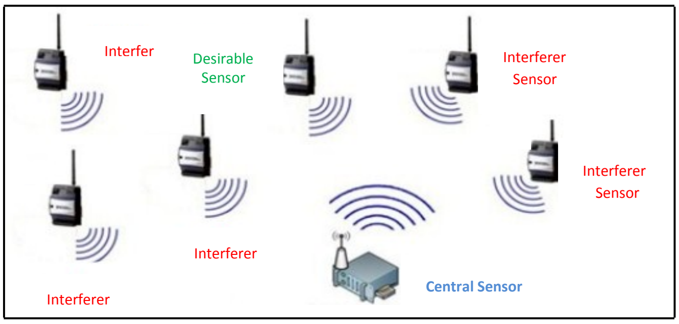

Fig.1. Wireless Sensor Network (WSN) with one desirable sensor, N interferer sensors and a central sensor.

At the receiver side (the central sensor), decoding every arrived signal may be wasteful of power since some receptions may involve colliding packets. Our proposed system distinguishes between one and more transmissions based on studying the statistics of received signal. Consequently, in building our system we consider two scenarios. The first scenario is when only one sensor is transmitting, and SNR (Signal to Noise Ratio at the receiver) is either better or worse than 5 dB (Minimum assumed SNR, where most coding schemes allow useful communication). SNR is the parameter that we consider and estimate in our algorithm and compare it against the minimum allowed SNR (5dB). If the SNR estimate is better than 5 dB, then our algorithm decides that there is only one sensor transmitting and the receiver must decode. If the SNR is worse than 5dB, then our algorithm confidently decide that there is more than one sensor transmitting and the receiver doesn't need to decode in order to save energy. The second scenario considers the interference case. Here we have one desirable sensor and a random number of interfering sensors. If the interference is dominant (i.e., more than the thermal noise) then we perform the same comparison of estimated SNR against the minimum allowed SNR (5 dB).

IV. ALGORITHM DESCRIPTION

Our algorithm is based upon evaluating the statistics of the received signal at the receiver ADC output via the use of a simple statistical discrimination metric calculation that is performed on a relatively small portion of the received IQ packet samples. The resulting metric value is then compared with a pre-specified threshold to determine if the statistics of the received packet samples reflect a signal-to-interference-plus-noise ratio (SINR) that is better than 5dB. If so, the packed is deemed sane (no collision) and qualifies for a full decoding procedure. Otherwise, the packet is deemed corrupt with other strong interferers (hence, a collision) and must be rejected without expending any further processing/decoding energy[1].

We consider four statistical discrimination metric formulations. A logarithmic (entropy) metric, a moment metric, maximum to minimum metric as follow:

$$Log_{metric} = \left(\frac{|log\ x_{noisy}|}{n}\right) \tag{1}$$

$$M_{metric} = \left(\frac{|(x_{noisy})^k|}{n}\right) \tag{2}$$

$$Max2Min_{metric} = \left(\frac{max\ |x_{noisy}|}{min\ |x_{noisy}|}/n\right) \tag{3}$$

where x_{noisy} is the transmitted signals over a noisy channels, k=3,5, or 7 is the moment's rank (or degree), e.g., third, fifth and seventh moments[2] .The metric is then compared with a pre-specified threshold that is set based on a 5dB SINR assumption. If the metric value reflects a SINR less than 5dB the packet is rejected. Hence, a "false alarm" scenario occurs if the metric erroneously deems the received SINR less than 5dB while it actually was higher than 5dB. On the other hand, if the metric deems the SINR to be higher than 5dB while it is actually less than 5dB, a "miss" scenario is encountered. Miss and False-Alarm probabilities have impact on the overall system throughput as will be discussed in the following sections. Therefore, it is required to minimize such probabilities as much as possible.

A. Threshold Selection

The decision threshold is chosen based on evaluating the false-alarm and miss probabilities and choosing the threshold values which satisfy the designer's requirements of such quantities. For example, we generate a 100,000 Monte-Carlo simulated snapshots of interfering sensors (e.g., 1~30 sensors with random received powers to simulate various path loss amounts) where for each snapshot we compute the discrimination metric value for the received total signal plus interference plus noise. In addition the simulator sweeps a range of threshold values for the snapshot at hand and determines if, for each threshold value, there would be an event of a false-alarm or a miss in order to count the probabilities of such events. At the end of the simulations the false-alarm and miss probabilities are plotted versus the range of evaluated threshold values which enables the designer to determine a satisfactory set point for the threshold.

V. POWER SAVING AND SYSTEM THROUGHPUT ANALYSIS

The simulation we built for our system is based on sending modulated signals from a desired sensor, these signals superimposed on a random number of interfering signals (coming from interferer sensors) plus AWGN noise component. Each sensor is assigned a random power level to simulate different path loss amounts. At the side of the central node, the sensors' power level and AWGN power are normalized in order to match the SNR requirements such as 3, 4.9, 5.1, 6dB. Figure 2 below is a flow chart for the simulator.

To analysis the power saving of our proposed system we introduce the following metrics:

$$F_{Bad} = (np)_{OurMetric} + P_{miss}\quad (np)_{ExistedMetric} \tag{4}$$

$$F_{Good} = (np)_{OurMetric} + (np)_{ExistedMetric} \tag{5}$$

$$C_{Bad} = (np)_{ExistedMetric} \tag{6}$$

$$C_{Good} = (np)_{ExistedMetric} \tag{7}$$

In above formulas, *np* is the number of computational operations incurred in our proposed metrics. P_{miss} is the probability of miss where the receiver make a wrong decision to fully decode the packet (declared as good packet). C_{Good} and C_{Good} are complex numbers used for power saving calculation.

Also we introduce the following formulas in order to compare the power saving of our proposed metrics with any other existing decoding metric:

$$T_F = F_{Bad}\quad P_{Bad}\ +\ F_{good}\quad P_{Good} \tag{8}$$

$$T_C = (C_{Bad} +\ C_{Good}\ )\ \ (np)_{ExistedMetric} \tag{9}$$

In above formulas, P_{Good} and P_{Bad} are the probability of good and bad codes, where the packet is declared as accepted and rejected packet respectively.

Finally, based on the calculations of false alarm probabilities, we figure out the throughput of our proposed system as follows:

$$\mathit{Throughput} = (1 - P_{FA})_{OurMetric} \tag{10}$$

A. Comparing with SOVA

The decoding of turbo codes can be divided into a maximum a posteriori (MAP) algorithm and Soft Output Viterbi Algorithm (SOVA). It is well known that the performance of MAP is superior in comparison with SOVA, however SOVA consumes less power (smaller number of addition and multiplication). Therefore, making more attractive for low power WSNs [21] [22]. The authors in [22] measure the complexity of the SOVA algorithm based

---

[1] A repeat request may be issued or the transmitting sensor may re-try depending on the MAC scheme, e.g., ALOHA.

[2] We have found that odd-valued moment ranks give better discrimination. Clearly, the second moment cannot be used as it represents the received signal power. Hence, it does not really bear any statistical discrimination information.

on the size of the encoder memory (M). The total amount of operations (Max-ops, additions, multiplication by + and – 1, bit comps) per bit the SOVA algorithm demands for decoding one code in one iteration is equal to 55, 76, 109 and 166.

As discussed above, our algorithm does not incur such computational complexity as SOVA and MAP algorithms. In addition, our algorithm avoids the other complexities required by a full decoding line-up such as time and frequency synchronization, Doppler shift correction, fading and channel estimation, etc., since our scheme operates directly at the IQ samples at the output of the ADC "as is" since it examines only the envelope of those IQ samples which is not statistically affected by such impairments. Finally, full deciding requires buffering and processing the entire packet (e.g., 1000 bits) while our scheme needs only to operate on a short portion of the received packet that could be as short as 25 bits as will be seen from the analytical results below.

In order to show the superior performance of our algorithm, we use the maximum to minimum metric as a case study and we assume estimation (test) duration of 50 bits. We assume a 8PSK modulation scheme. We compare with the SOVA algorithm [22] where M=5 and we assume for the sake of argument that the entire packet length is equal to the test period which is 50 bits. Figure 2 shows the probability of miss and false alarm are equal to 22.70% for a SINR that is within 1.5 dB from the cut-off SINR (5dB).

Now, based on equations (4) and (5), F_{Bad} and F_{Good} of our algorithm will be equal:

$$F_{Bad} = (np)_{OurMetric} + P_{miss}\ (np)_{ExistMetric}$$

$$= (64)_{OurMetric} + 22.50\%\ \ (166)_{ExistMetric}$$

$$= 101.35$$

$$F_{Good} = (np)_{OurMetric} + (np)_{ExistedMetric}$$

$$= 64 + 166$$

$$= 230$$

In addition, C_{Bad}, C_{good} and T_C in equations (6), (7) and (9) will be equal 166. Our algorithm power saving based on equation (8) is equal to:

$$T_F = F_{Bad}\ \ P_{Bad}\ +\ F_{good}\ \ P_{Good}$$

$$= (101.35)\ (63.51\ \%) + (230)\ (36.49\%)$$

$$= 148.29$$

The system throughput as defined in equation (10) is then equal to:

$$Throughput = (1 - P_{FA})_{OurMetric} = 1\text{- } 22.50\ \% = 77.50\ \%$$

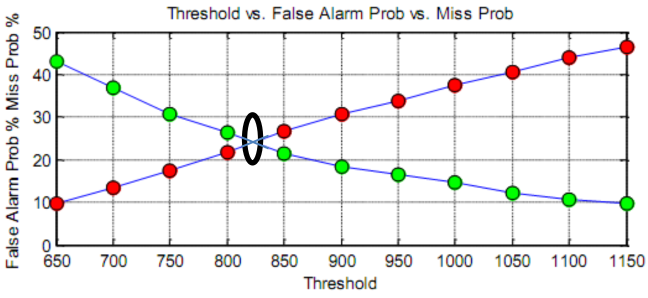

Fig.2. The intersection between false alarm bribability = 22.50% and miss probability = 22.50% at threshold point = 825, 8PSK.

Note that the above performance examples can be tuned as desired. The system designer may choose to reduce the number of transmitted bits (25 or 50 bits) at the expense of increasing the miss and false alarm probabilities, or may increase the throughput by using a longer estimation period in order to improve the accuracy of the statistical metric performance and reduce the miss and false–alarm probabilities.

VI. RESULTS AND DISCUSSION

We have generated 100,000 simulation snapshots where each snapshot generates a random number of sensors up to 30 sensors with random power assignments (or equivalently path loss, i.e., assignments).

All proposed metrics exhibit robust performance. In our study, we have evaluated various MPSK modulation schemes (e.g. QPSK, 8PSK and 16PSK) versus various measurement durations, sampling rates and metric numerical (fixed-point) quantization levels to reflect the effects of practical implementation constraints.

Our proposed algorithm has a low sensitivity to deviations of the received SINR from the assumed set-point which is 5dB. The algorithm works reliably and able to determine if the packet is in collision or not. That is if the SINR is well below or above the set-point, the received signal statistics are expected to also be less confusing to the discriminator anyway and the algorithm shall perform reliably.

Figures 3 and 4 show the miss and false-alarm probabilities versus the choice of the metric comparison threshold level (i.e., above which we decide the packet is in collision or not) for the proposed metrics for QPSK, 8PSK and 16PSK modulation schemes.

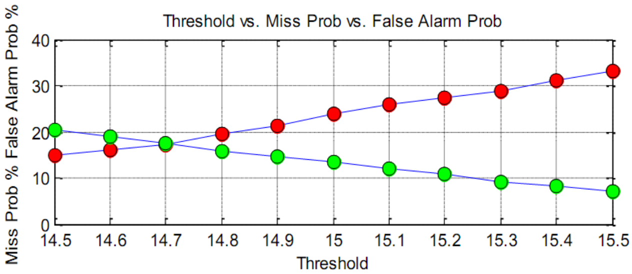

Fig.3. Miss probability =18.33% vs. false alarm probability=17.36% vs. threshold=14.7, SNR=1dB up/below cutoff SNR=5dB, logarithm metric, QPSK, NumBit_Log=8 bits, sampling rate=6, transmitted signal= 50 bits.

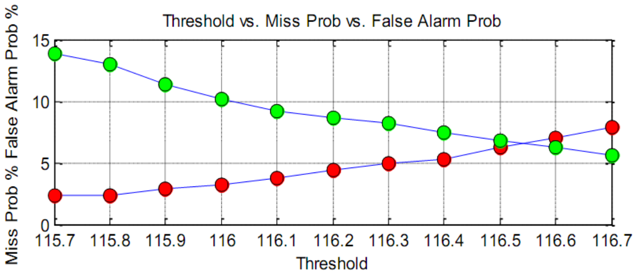

Fig.4. Miss probability = 7.27% vs. false alarm probability=6.85% vs. threshold=116.5, SNR=1dB up/below cutoff SNR=5dB, 3rd moment metric, 8PSK, NumBit_Log=10 bits, sampling rate=8, transmitted signal= 200 bits.

In general, in order to have a fair treatment for miss and false alarm probabilities, a designer can choose an arbitrarily different threshold level. Thus, the associated figures and Appendix (for the maximum to minimum metric) can be regarded as guide for a designer. For example, in figure 5 if a designer chooses a threshold level 1900, the probability of false alarm will be 7.2% which is a reasonable probability. That is, the algorithm with a low false alarm probability can correctly determine the packet is in collision and hence it needs to be rejected. Also, in figure 6 when the choice of a threshold level is 1500, the probability of miss will be 4.1% (i.e. the packet isn't in collision and it must follow a full decoding procedure). Low false alarm probability has impact on the overall system throughput. The system throughput increases with decrease in false alarm probability as discussed in the previous section.

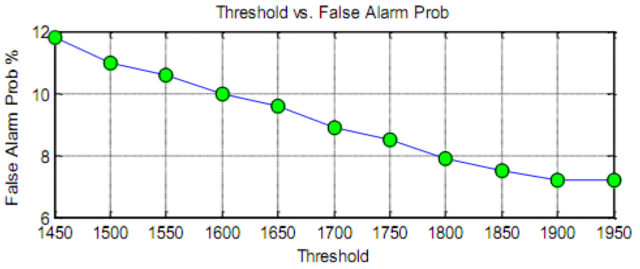

Fig.5. False alarm probabilities vs. threshold, SNR=1.5dB up/below cutoff SNR=5dB, Max2Min metric, QPSK, 50 bits transmitted.

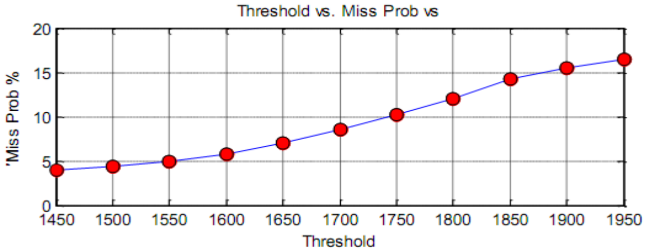

Fig.6. Miss probabilities vs. threshold, SNR=1.5dB up/below cutoff SNR=5dB , Max2Min metric, 8PSK, 200 bits transmitted

VII. CONCLUSION

In this paper we analyze the performance of a novel power saving algorithm for WSNs. Our proposed algorithm is based on studying the statistics of received signals and hence the receiver can make a fast decision to decode or reject a packet. Our proposed algorithm is based on three simple discrimination metrics which have low computational complexities as well as short measurement period requirements. In addition, our proposed algorithm minimize the delay when decoding the received packet, it analyses the sanity of small portion of the received packet instead of waiting the entire packet to be buffered/received in order to start decoding (the case in most decoding systems). The analysis and associated figures/tables presented in this paper can be regarded as a designer's guide for achieving significant power saving with low-complexity and low throughput loss.

APPENDIX

TABLES FOR SIMULATION RESULTS

TABLE I
QPSK – LOGARITH METRIC

Logarithm Metric							
QPSK							
N NumBit_ Arg	**M NumBit_ Log**	**S_Rate**	**SNR OFF in dB**	**FA Prob**	**Miss Prob**	**Thrsh_ Point**	**No # Samples**
25	4	2	1	33.33%	32.88%	14.7	25
25	6	4	1	29.44%	29.92%	14.9	50
25	8	6	1	26.90%	27.11%	14.9	75
25	10	8	1	24.98%	26.02%	15.0	100
25	4	2	1.5	26.47%	25.79%	14.6	25
25	6	4	1.5	21.11%	20.37%	14.8	50
25	8	6	1.5	18.41%	18.48%	14.9	75
25	10	8	1.5	16.27%	17.11%	15.0	100
50	4	2	1	24.72%	25.60%	14.6	50
50	6	4	1	19.71%	20.97%	14.7	100
50	8	6	1	17.36%	18.33%	14.7	150
50	10	8	1	15.97%	16.23%	14.7	200
50	4	2	1.5	16.24%	15.76%	14.4	50
50	6	4	1.5	9.99%	10.64%	14.6	100
50	8	6	1.5	8.12%	7.97%	14.6	150
50	10	8	1.5	6.84%	7.62%	14.6	200
200	4	2	1	9.10%	8.44%	14.2	200
200	6	4	1	4.61%	3.96%	14.2	400
200	8	6	1	3.03%	3.24%	14.3	600
200	10	8	1	2.58%	2.06%	`14.3	800
200	4	2	1.5	1.93%	2.02%	14.1	200
200	6	4	1.5	0.47%	0.56%	14.2	400
200	8	6	1.5	0.18%	0.26%	14.3	600
200	10	8	1.5	0.16%	0.14%	14.2	800
500	4	2	1	1.96%	1.99%	14.1	500
500	6	4	1	0.56%	0.41%	14.1	1000
500	8	6	1	0.35%	0.26%	14.2	1500
500	10	8	1	0.20%	0.15%	14.2	2000
500	4	2	1.5	0.11%	0.09%	14.1	500
500	6	4	1.5	0.00%	0.00%	14.2	1000
500	8	6	1.5	0.00%	0.00%	14.1	1500
500	10	8	1.5	0.00%	0.00%	14.2	2000
1000	4	2	1	0.25%	0.52%	14.1	1000
1000	6	4	1	0.00%	0.00%	14.1	2000
1000	8	6	1	0.00%	0.00%	14.1	3000
1000	10	8	1	0.00%	0.00%	14.1	4000
1000	4	2	1.5	0.00%	0.00%	14.1	1000
1000	6	4	1.5	0.00%	0.00%	14.1	2000
1000	8	6	1.5	0.00%	0.00%	14.1	3000
1000	10	8	1.5	0.00%	0.00%	14.1	4000

TABLE II
8PSK – LOGARITH METRIC

Logarithm Metric							
8PSK							
N NumBit_ Arg	**M NumBit_ Log**	**S_ Rate**	**SNR OFF in dB**	**FA_ Prob**	**Miss_ Prob**	**Thrsh_ Point**	**No # Samples**
25	4	2	1	36.77%	36.71%	14.5	16
25	6	4	1	33.62%	33.49%	14.8	32
25	8	6	1	31.94%	30.72%	14.8	48
25	10	8	1	29.56%	29.81%	14.9	64
25	4	2	1.5	30.90%	30.08%	14.3	16
25	6	4	1.5	25.62%	25.60%	14.7	32
25	8	6	1.5	23.53%	23.17%	14.8	48
25	10	8	1.5	22.52%	21.62%	14.9	64
50	4	2	1	28.72%	29.63%	14.7	32
50	6	4	1	24.87%	25.49%	14.9	64
50	8	6	1	23.38%	23.64%	14.9	96
50	10	8	1	22.25%	22.28%	14.9	128
50	4	2	1.5	22.23%	22.46%	14.6	32
50	6	4	1.5	16.39%	16.67%	14.8	64
50	8	6	1.5	13.37%	13.66%	14.8	96
50	10	8	1.5	12.38%	12.08%	14.8	128
200	4	2	1	12.60%	13.01%	14.3	132
200	6	4	1	7.62%	7.59%	14.3	264
200	8	6	1	6.18%	5.55%	14.3	396
200	10	8	1	4.50%	5.65%	`14.4	627
200	4	2	1.	4.85%	4.54%	14.1	132
200	6	4	1.5	1.76 %	1.67 %	14.2	264
200	8	6	1.5	0.82 %	0.97%	14.3	396
200	10	8	1.5	0.77%	0.63%	14.3	627
500	4	2	1	4.00%	3.87%	14.1	332
500	6	4	1	1.04%	1.63%	14.2	664

500	8	6	1	0.69%	0.84%	14.2	996
500	10	8	1	0.53%	0.53%	14.2	1328
500	4	2	1.5	0.46%	0.36%	13.9	332
500	6	4	1.5	0.05%	0.05%	14.0	664
500	8	6	1.5	0.00%	0.00%	14.2	996
500	10	8	1.5	0.00%	0.00%	14.2	1328
1000	4	2	1	0.74%	0.94%	14.1	666
1000	6	4	1	0.00%	0.00%	14.1	1332
1000	8	6	1	0.00%	0.00%	14.1	1998
1000	10	8	1	0.00%	0.00%	14.1	2664
1000	4	2	1.5	0.00%	0.00%	14.0	666
1000	6	4	1.5	0.00%	0.00%	14.0	1332
1000	8	6	1.5	0.00%	0.00%	14.0	1998
1000	10	8	1.5	0.00%	0.00%	14.0	2664

TABLE III
16PSK – LOGARITH METRIC

Logarithm Metric							
16PSK							
N NumBit Arg	**M NumBit Log**	**S_Rate**	**SNR OFF in dB**	**FA_ Prob**	**Miss_ Prob**	**Thrsh_ Point**	**No # Samples**
25	4	2	1	38.61%	39.45%	13.9	12
25	6	4	1	35.49%	35.49%	14.3	24
25	8	6	1	33.86%	33.84%	14.4	36
25	10	8	1	31.33%	31.33%	14.4	48
25	4	2	1.5	33.24%	32.76%	13.8	12
25	6	4	1.5	28.70 %	29.25%	14.3	24
25	8	6	1.5	23.04%	27.34%	14.4	36
25	10	8	1.5	24.58 %	24.94%	14.5	48
50	4	2	1	33.14%	33.04%	14.7	24
50	6	4	1	28.83%	30.06%	15.0	48
50	8	6	1	26..34%	27.51%	15.0	72
50	10	8	1	25.85%	24.54%	14.9	96
50	4	2	1.5	26.21%	25.80%	14.6	24
50	6	4	1.5	20.65%	20.65%	14.8	48
50	8	6	1.5	18.29%	18.43%	14.9	72
50	10	8	1.5	16.77 %	15.71 %	14.9	96
200	4	2	1	16.92%	16.86%	14.3	100
200	6	4	1	11.26%	10.11%	14.3	200
200	8	6	1	8.10%	8.44%	14.4	300
200	10	8	1	7.68%	6.67%	`14.4	400
200	4	2	1.5	7.10%	6.92%	14.2	100
200	6	4	1.5	3.35%	2.85%	14.2	200
200	8	6	1.5	2.00%	1.86%	14.3	300
200	10	8	1.5	1.43	1.75%	14.4	400
500	4	2	1	5.27%	6.48%	14.2	250
500	6	4	1	2.25%	2.73%	14.2	500
500	8	6	1	1.71%	1.56%	14.2	750
500	10	8	1	0.99%	1.39%	14.3	1000
500	4	2	1.5	1.00%	1.22%	14.0	250
500	6	4	1.5	0.15%	0.24%	14.1	500
500	8	6	1.5	0.07%	0.11%	14.2	750
500	10	8	1.5	0.05%	0.03%	14.2	1000
1000	4	2	1	1.44%	2.03%	14.1	500
1000	6	4	1	0.38%	0.30%	14.1	1000
1000	8	6	1	0.00%	0.00%	14.1	1500
1000	10	8	1	0.00%	0.00%	14.1	2000
1000	4	2	1.5	0.00%	0.00%	14.0	500
1000	6	4	1.5	0.00%	0.00%	14.0	1000
1000	8	6	1.5	0.00%	0.00%	14.0	1500
1000	10	8	1.5	0.00%	0.00%	14.0	2000

TABLE IV
QPSK – 3RD MOMENT METRIC

3rd Moment Metric							
QPSK							
N NumBit_ Arg	**M NumBit_ Mom**	**S_Rate**	**SNR OFF in dB**	**FA Prob**	**Miss Prob**	**Thrsh_ Point**	**No # Samples**
25	4	2	1	33.55%	34.10%	116.4	25
25	6	4	1	30.36%	30.01%	116.9	50
25	8	6	1	29.26%	28.04%	117.0	75
25	10	8	1	27.31%	26.86%	117.0	100
25	4	2	1.5	26.88%	26.99%	116.2	25
25	6	4	1.5	22.31%	22.62%	116.8	50
25	8	6	1.5	20.05%	20.94%	117.0	75
25	10	8	1.5	18.80%	18.86%	117.1	100
50	4	2	1	27.87%	27.77%	116.7	50
50	6	4	1	23.11%	24.08%	117.0	100
50	8	6	1	23.52%	22.67%	116.9	150
50	10	8	1	21.20%	20.97%	117.0	200
50	4	2	1.5	19.05%	19.16%	116.5	50
50	6	4	1.5	14.50%	14.35%	116.9	100
50	8	6	1.5	11.73%	12.16%	117.0	150
50	10	8	1.5	11.35%	10.97%	117.1	200
200	4	2	1	10.54%	10.91%	116.0	200
200	6	4	1	6.77%	6.88%	116.1	400
200	8	6	1	6.06%	5.50%	116.1	600
200	10	8	1	5.15%	5.29%	116.2	800
200	4	2	1.5	3.57%	3.57%	115.8	200
200	6	4	1.5	1.59%	1.83%	116.0	400
200	8	6	1.5	1.45%	1.35%	115.9	600
200	10	8	1.5	1.44%	1.36%	116.0	800
500	4	2	1	3.33%	3.33%	115.6	500
500	6	4	1	1.47%	1.57%	115.6	1000
500	8	6	1	1.17%	1.24%	115.6	1500
500	10	8	1	1.12%	1.22%	115.5	2000
500	4	2	1.5	0.48%	0.43%	115.2	500
500	6	4	1.5	0.16%	0.15%	115.3	1000
500	8	6	1.5	0.15%	0,17%	115.2	1500
500	10	8	1.5	0.14%	0.16%	115.1	2000
1000	4	2	1	0.78%	0.84%	115.4	1000
1000	6	4	1	0.07%	0.08%	115.4	2000
1000	8	6	1	0.00%	0.00%	115.4	3000
1000	10	8	1	0.00%	0.00%	115.4	4000
1000	4	2	1.5	0.00%	0.00%	115.2	1000
1000	6	4	1.5	0.00%	0.00%	115.2	2000
1000	8	6	1.5	0.00%	0.00%	115.2	3000
1000	10	8	1.5	0.00%	0.00%	115.2	4000

TABLE V
8PSK – 3RD MOMENT METRIC

3rd Moment Metric							
8PSK							
N NumBit_ Arg	**M NumBit_ Mom**	**S_Rate**	**SNR OFF in dB**	**FA Prob**	**Miss Prob**	**Thrsh_ Point**	**No # Samples**
25	4	2	1	37.16%	37.16%	115.4	16
25	6	4	1	33.49%	33.00%	116.2	32
25	8	6	1	30.89%	30.81%	116.4	48
25	10	8	1	29.80%	30.31%	116.5	64
25	4	2	1.5	30.30%	30.65%	115.4	16
25	6	4	1.5	26.42%	25.37%	116.0	32
25	8	6	1.5	22.60%	22.97%	116.3	48
25	10	8	1.5	22.37%	21.80%	116.4	64
50	4	2	1	31.76%	31.05%	116.7	32
50	6	4	1	27.97%	27.88%	117.1	64
50	8	6	1	25.81%	26.53%	117.2	96
50	10	8	1	24.91%	24.43%	117.2	128
50	4	2	1.5	24.06%	24.56%	116.6	32
50	6	4	1.5	19.39%	18.88%	117.0	64
50	8	6	1.5	17.20%	17.29%	117.2	96
50	10	8	1.5	15.74%	15.39%	117.2	128
200	4	2	1	16.30%	15.56%	116.2	132
200	6	4	1	11.25%	11.30%	116.4	264
200	8	6	1	9.22%	9.46%	116.4	396
200	10	8	1	6.85%	7.27%	116.5	627
200	4	2	1.5	6.10%	6.19%	116.1	132
200	6	4	1.5	4.01%	3.84%	116.3	264
200	8	6	1.5	3.48%	3.27%	116.3	396
200	10	8	1.5	3.04%	2.79%	116.4	627
500	4	2	1	5.57%	5.75%	115.8	332
500	6	4	1	3.26%	3.07%	115.8	664
500	8	6	1	2.44%	2.34%	115.8	996
500	10	8	1	2.14%	2.27%	115.8	1328
500	4	2	1.5	0.95%	1.15%	115.6	332
500	6	4	1.5	0.56%	0.53%	115.5	664
500	8	6	1.5	0.43%	0.45%	115.6	996
500	10	8	1.5	0.41%	0.36%	115.4	1328
1000	4	2	1	1.69%	1.69%	115.5	666
1000	6	4	1	0.79%	0.86%	115.4	1332
1000	8	6	1	0.64 %	0.75%	115.4	1998
1000	10	8	1	0.56%	0.56%	115.4	2664
1000	4	2	1.5	0.19%	0.17%	115.2	666
1000	6	4	1.5	0.05%	0.02%	115.0	1332
1000	8	6	1.5	0.00%	0.00%	115.2	1998
1000	10	8	1.5	0.00%	0.00%	115.2	2664

TABLE VI
16PSK – 3RD MOMENT METRIC

3rd Moment Metric							
16PSK							
N NumBit_ Arg	**M NumBit_ Mom**	**S_Rate**	**SNR OFF in dB**	**FA Prob**	**Miss Prob**	**Thrsh_ Point**	**No # Samples**
25	4	2	1	38.37%	38.89%	114.1	12
25	6	4	1	35.08%	34.92	115.0	24
25	8	6	1	34.09%	33.37%	115.3	36
25	10	8	1	31.57%	31.92%	115.6	48
25	4	2	1.5	33.06%	33.38%	114.0	12
25	6	4	1.5	28.26%	28.58%	114.9	24
25	8	6	1.5	25.93%	25.75%	115.3	36
25	10	8	1.5	24.22%	24.44%	115.4	48
50	4	2	1	33.71%	34.13%	116.6	24
50	6	4	1	31.16%	30.80%	117.0	48
50	8	6	1	28.37%	28.07%	117.0	72
50	10	8	1	27.59%	26.89%	117.1	96
50	4	2	1.5	26.99%	27.58%	116.3	24
50	6	4	1.5	21.89%	21.99%	116.9	48
50	8	6	1.5	19.57%	20.01%	117.1	72
50	10	8	1.5	18.90%	19.28%	117.2	96
200	4	2	1	18.92%	19.78%	116.4	100
200	6	4	1	13.99%	14.55%	116.5	200
200	8	6	1	13.53%	12.73%	116.6	300
200	10	8	1	11.30%	12.14%	116.7	400
200	4	2	1.5	9.59%	9.59%	116.3	100
200	6	4	1.5	6.40%	6.40%	116.5	200
200	8	6	1.5	5.29%	4.91%	116.5	300
200	10	8	1.5	4.33%	4.22%	116.6	400
500	4	2	1	7.79%	8.51%	115.9	250
500	6	4	1	4.84%	5.12%	116.0	500
500	8	6	1	3.71%	4.30%	116.0	750
500	10	8	1	3.48%	3.29%	116.0	1000
500	4	2	1.5	1.98%	1.98%	115.7	250
500	6	4	1.5	1.01%	1.09%	115.8	500
500	8	6	1.5	0.94%	0.81%	115.8	750
500	10	8	1.5	0.81%	0.78%	115.8	1000
1000	4	2	1	3.00%	3.08%	115.6	500
1000	6	4	1	1.50%	1.50%	115.6	1000
1000	8	6	1	1.33%	1.10%	115.5	1500
1000	10	8	1	1.07%	1.17%	115.5	2000
1000	4	2	1.5	0.37%	0.40%	115.4	500
1000	6	4	1.5	0.24%	0.20%	115.2	1000
1000	8	6	1.5	0.21%	0.18%	115.1	1500
1000	10	8	1.5	0.11%	0.13%	115.2	2000

TABLE VII
QPSK – MAXIMUM TO MINIMUM METRIC

Max2Min Metric							
QPSK							
N NumBit Arg	**M NumBit Log**	**S Rate**	**SNR OFF in dB**	**FA Prob**	**Miss Prob**	**Thrsh Point**	**No # Samples**
25	4	2	1	36.33%	35.23%	550	25
25	6	4	1	34.44%	33.92%	750	50
25	8	6	1	29.90%	30.11%	950	75
25	10	8	1	24.98%	26.02%	1100	150
25	4	2	1.5	31.47%	30.79%	550	25
25	6	4	1.5	27.11%	24.82%	750	50
25	8	6	1.5	25.41%	23.90%	900	75
25	10	8	1.5	21.37%	22.11%	1050	150

50	4	2	1	33.52%	34.60%	800	50
50	6	4	1	29.30%	31.97%	1100	100
50	8	6	1	28.36%	30.33%	1300	150
50	10	8	1	26.97%	27.23%	1550	200
50	4	2	1.5	27.24%	28.76%	1350	50
50	6	4	1.5	24.74%	23.64%	1300	100
50	8	6	1.5	23.12%	22.65%	1100	150
50	10	8	1.5	20.14%	21.01%	1050	200
200	4	2	1	28.10%	27.44%	1450	200
200	6	4	1	25.34%	23.21%	1500	400
200	8	6	1	24.81%	22.01%	1500	600
200	10	8	1	18.34%	18.93%	`1550	800
200	4	2	1.5	17.34%	19.84%	1450	200
200	6	4	1.5	15.73%	17.83 %	1500	400
200	8	6	1.5	13.23%	15.34%	1500	600
200	10	8	1.5	11.54%	12.34%	1500	800
500	4	2	1	22.21%	23.43%	1550	500
500	6	4	1	18.24%	19.34%	1550	1000
500	8	6	1	16.26%	15.34%	1600	1500
500	10	8	1	12.20%	12.15%	1600	2000
500	4	2	1.5	14.65%	15.09%	1550	500
500	6	4	1.5	12.00%	12.30%	1550	1000
500	8	6	1.5	10.40%	10.60%	1650	1500
500	10	8	1.5	8.34%	8.90%	1650	2000
1000	4	2	1	20.65%	19.32%	1700	1000
1000	6	4	1	18.34%	17.89%	1750	2000
1000	8	6	1	15.55%	15.34%	1750	3000
1000	10	8	1	11.78%	11.34%	1750	4000
1000	4	2	1.5	12.32%	13.34%	1700	1000
1000	6	4	1.5	9.90%	9.10%	1800	2000
1000	8	6	1.5	7.45%	7.23%	1850	3000
1000	10	8	1.5	5.93%	5.55%	1850	4000

TABLE VIII
8PSK – MAXIMUM TO MINIMUM METRIC

Max2Min Metric							
8PSK							
N NumBit Arg	**M NumBit Log**	**S Rate**	**SNR OFF in dB**	**FA Prob**	**Miss Prob**	**Thrsh point**	**No # Samples**
25	4	2	1	39.07%	39.71%	450	16
25	6	4	1	36.62%	35.49%	450	32
25	8	6	1	33.94%	32.72%	450	48
25	10	8	1	30.46%	30.81%	500	64
25	4	2	1.5	32.70%	31.03%	450	16
25	6	4	1.5	29.62%	29.60%	500	32
25	8	6	1.5	27.53%	25.17%	550	48
25	10	8	1.5	24.52%	23.62%	550	64
50	4	2	1	34.72%	35.63%	650	32
50	6	4	1	30.87%	33.45%	650	64
50	8	6	1	29.38%	31.64%	650	96
50	10	8	1	28.25%	28.28%	700	128
50	4	2	1.5	29.33%	29.49%	600	32
50	6	4	1.5	25.39%	24.67%	600	64
50	8	6	1.5	24.37%	23.66%	600	96
50	10	8	1.5	23.34%	22.03%	700	128
200	4	2	1	31.20%	29.01%	1200	132
200	6	4	1	30.01%	28.12%	1200	264
200	8	6	1	28.91%	26.12%	1200	396
200	10	8	1	26.22%	24.66%	`1250	627
200	4	2	1.	21.01%	22.22%	1200	132
200	6	4	1.5	19.17%	19.10%	1200	264
200	8	6	1.5	17.33%	17.12%	1350	396
200	10	8	1.5	15.23%	15.74%	1350	627
500	4	2	1	25.37%	26.34%	1900	332
500	6	4	1	23.33%	24.15%	1900	664
500	8	6	1	21.91%	22.45%	1950	996
500	10	8	1	20.17%	21.84%	1950	1328
500	4	2	1.5	21.45%	19.12%	1800	332
500	6	4	1.5	18.26%	17.21%	1800	664
500	8	6	1.5	16.14%	14.22%	1800	996
500	10	8	1.5	14.87%	11.67%	1850	1328
1000	4	2	1	27.23%	26.33%	2750	666
1000	6	4	1	24.45 %	23.32%	2700	1332
1000	8	6	1	20.12%	22.34%	2700	1998
1000	10	8	1	19.65%	20.32%	2700	2664
1000	4	2	1.5	21.00%	21.00%	2750	666
1000	6	4	1.5	18.35%	15.45%	2750	1332
1000	8	6	1.5	14.76%	12.35%	2800	1998
1000	10	8	1.5	10.67%	9.38%	2800	2664

TABLE IX
16PSK – MAXIMUM TO MINIMUM METRIC

Max2Min Metric							
16PSK							
N NumBit Arg	**M NumBit Log**	**S Rate**	**SNR OFF in dB**	**FA Prob**	**Miss Prob**	**Thrsh Point**	**No # Samples**
25	4	2	1	41.07%	40.73%	400	12
25	6	4	1	39.62%	37.44%	400	24
25	8	6	1	35.92%	33.76%	400	36
25	10	8	1	31.49%	31.86%	500	48
25	4	2	1.5	35.71%	33.07%	400	12
25	6	4	1.5	30.62%	30.65%	400	24
25	8	6	1.5	29.03%	27.37%	550	36
25	10	8	1.5	26.50%	25.62%	550	48
50	4	2	1	36.02%	37.73%	650	24
50	6	4	1	33.82%	35.45%	650	48
50	8	6	1	32.36%	33.64%	750	72
50	10	8	1	31.21%	30.83%	750	96
50	4	2	1.5	30.36%	29.49%	800	24
50	6	4	1.5	27.29%	25.64%	800	48
50	8	6	1.5	26.37%	24.66%	800	72
50	10	8	1.5	25.34%	23.03%	800	96
200	4	2	1	33.10%	30.01%	1200	100
200	6	4	1	31.01%	29.12%	1200	200
200	8	6	1	29.91%	27.12%	1250	300
200	10	8	1	27.22%	26.69%	`1250	400
200	4	2	1.	23.01%	24.22%	1200	100
200	6	4	1.5	20.17%	20.30%	1250	200
200	8	6	1.5	19.33%	18.12%	1300	300
200	10	8	1.5	16.83%	16.79%	1350	400
500	4	2	1	27.37%	27.34%	1900	250
500	6	4	1	24.63%	26.85%	1950	500
500	8	6	1	23.91%	24.35%	1950	750
500	10	8	1	21.07%	22.14%	1950	1000
500	4	2	1.5	22.95%	20.12%	1800	250
500	6	4	1.5	19.20%	18.21%	1900	500
500	8	6	1.5	17.24%	16.81%	1900	750
500	10	8	1.5	15.85%	13.67%	1950	1000
1000	4	2	1	28.20%	27.73%	2300	500
1000	6	4	1	26.05%	25.31%	2300	1000
1000	8	6	1	22.42%	23.24%	2300	1500
1000	10	8	1	20.95%	21.32%	2350	2000
1000	4	2	1.5	23.03%	23.00%	2300	500
1000	6	4	1.5	19.38%	16.43%	2300	1000
1000	8	6	1.5	15.76%	13.35%	2350	1500
1000	10	8	1.5	11.61%	10.91%	2350	2000

REFRENCES

[1] P. Minet, “Energy efficient routing”, in *Ad Hoc and Sensor Wireless Networks: Architectures: Algorithms and Protocols*. Bentham Science, 2009.

[2] Soua, R.; Minet, P., "A survey on energy efficient techniques in wireless sensor networks," Wireless and Mobile Networking Conference (WMNC), 2011 4th Joint IFIP , vol., no., pp.1,9, 26-28 Oct. 2011.

[3] Mahfoudh, S.; Minet, P., "Survey of Energy Efficient Strategies in Wireless Ad Hoc and Sensor Networks," Networking, 2008. ICN 2008. Seventh International Conference on , vol., no., pp.1,7, 13-18 April 2008.

[4] Damaso, Antonio; Freitas, Davi; Rosa, Nelson; Silva, Bruno, Paulo, "Evaluating the Power Consumption of Wireless Sensor Network Application Using Models." Sensors 2013, 13, 3473-355.

[5] Sandra, S.; Jaime, L.; Miguel, G.; Jose, F.T, "Power Saving and Energy Optimization Techniques for Wireless Sensor Networks (Invited Paper)." Journal of Communications, vol. 6. No.6, pp.439-459, 2011.

[6] Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H., "Energy-efficient communication protocol for wireless microsensor networks," System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on , vol., no., pp.10 pp. vol.2,, 4-7 Jan. 2000.

[7] Ingelrest, F.; Simplot-Ryl, D.; Stojmenovic, I., "Smaller Connected Dominating Sets in Ad Hoc and Sensor Networks based on Coverage by Two-Hop Neighbors," Communication Systems Software and Middleware, 2007. COMSWARE 2007. 2nd International Conference on , vol., no., pp.1,8, 7-12 Jan. 2007.

[8] Kimura, N.; Latifi, S., "A survey on data compression in wireless sensor networks," Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on , vol.2, no., pp.8,13 Vol. 2, 4-6 April 2005.

[9] Li, N.; Hou, J.C.; LuiSha, "Design and analysis of an MST-based topology control algorithm," Wireless Communications, IEEE Transactions on , vol.4, no.3, pp.1195,1206, May 2005.

[10] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, J. A. Stankovic, ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks, SenSys’06,Colorado, November 2006.

[11] Sungoh Kwon; Shroff, N.B., "Energy-Efficient Interference-Based Routing for Multi-Hop Wireless Networks," INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings , vol., no., pp.1,12, April 2006.

[12] da S Araújo, H.; HolandaFilho, R.; Filho, R.H., "WSN Routing: An Geocast Approach for Reducing Consumption Energy," Wireless Communications and Networking Conference (WCNC), 2010 IEEE , vol., no., pp.1,6, 18-21 April 2010.

[13] El-Aaasser, M.; Ashour, M., "Energy aware classification for wireless sensor networks routing," Advanced Communication Technology (ICACT), 2013 15th International Conference on , vol., no., pp.66,71, 27-30 Jan. 2013.

[14] Cardei, M.; Thai, M.T.; Yingshu Li; Weili Wu, "Energy-efficient target coverage in wireless sensor networks," INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE , vol.3, no., pp.1976,1984 vol. 3, 13-17 March 2005.

[15] Junchao Ma; Wei Lou; Yanwei Wu; Mo Li; Guihai Chen, "Energy Efficient TDMA Sleep Scheduling in Wireless Sensor Networks," INFOCOM 2009, IEEE , vol., no., pp.630,638, 19-25 April 2009.

[16] Sami H. O. Salih andMamoun M. A. Suliman, "Implementation of AMC Technique" International Journal of Scientific & Engineering Research Volume 2, Issue 5, May-2011, ISSN 2229-5518,2011.

[17] Mohammad Rakibul Islam, "Error Correction Codes in Wireless SensorNetwork: An Energy Aware approach" International Journal of Computer and Information Engineering 4:1, 2010.

[18] GopinathBalakrishnan, M. Yang, Y. Jiang, and Y. Kim, "Performance analysis of error control codes for wireless sensor networks", International Conference on Information Technology (ITNG'07), pp. 876–879,2007.

[19] Mukesh, S.; Iqbal, M.; Zhang Jianhua; Zhang Ping; Inam-Ur-Rehman, "Comparative Analysis of M-ary Modulation Techniques for Wireless Ad-hoc Networks," Sensors Applications Symposium, 2007. SAS '07. IEEE , vol., no., pp.1,6, 6-8 Feb. 2007.

[20] Sheryl L. Howard, Christian Schlegel, and Kris Iniewski, "Error Control Coding in Low-Power Wireless Sensor Networks: When Is ECC Energy Efficient?" EURASIP Journal on Wireless Communications and Networking,Volume 2006, Article ID 74812, Pages 1–14DOI 10.1155/WCN/2006/74812,2006.

[21] Woo Tae Kim; Sang Jae Bae; Seog Geun Kang; Eon Kyeong Joo, "Reduction of computational complexity in two-step SOVA decoder for turbo code," Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE , vol.3, no., pp.1887,1891 vol.3, 2000.

[22] Robertson, P.; Villebrun, E.; Hoeher, P., "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," Communications, 1995. ICC '95 Seattle, 'Gateway to Globalization', IEEE International Conference on , vol.2, no., pp.1009,1013 vol.2, Jun 1995.