Energy Efficient Transmission Error Recovery for Wireless Sensor Networks

Shio Kumar Singh [1], M P Singh [2], and D K Singh [3]

[1]Maintenance Engineering Department (Electrical), Tata Steel Limited, Jamshedpur – 831001, Jharkhand, India, shio.singh@tatasteel.com
[2] Computer Science and Engineering Department, National Institute of Technology, Patna, Bihar, India, writetomps@gmail.com
[3] Electronics and Communication Engineering Department, Birsa Institute of Technology, Sindri, Dhanbad –828123, Jharkhand, India, dkdingh_bit@yahoo.com

Abstract

Transmission errors are inherent in wireless communications because of the instability of wireless channels resulting due to channel fading, time-frequency coherence, inter-band interference, etc, and thus receiving unreliable message packet. This necessitates the retransmission of packet which further results into extra processing and increased energy consumption of the sensor nodes.

In this paper, we propose a new approach of an energy-efficient transmission error recovery algorithm for wireless sensor networks. In the proposed methodology, the data packet is divided into small sub-packets and retransmission is done for only corrupted portion of the sub-packets. The algorithm also helps in determining optimum size of the sub-packet. These results into minimum battery power requirement for the sensor nodes and increased lifespan of the sensor network. The results show that more than 30% of the power savings can be achieved by implementing proposed system as compared to conventional retransmission of complete packet.

Keywords: *Wireless Sensor Networks (WSNs), Forward Error Correction (FEC), Automatic Repeat Request (ARQ)*

1. Introduction

Wireless sensor network (WSN) is an emerging technology that promises a wide range of potential applications in both civilian and military areas [10], for example, environmental monitoring, battlefield monitoring and remote sensing, healthcare applications, industrial process control, security and surveillance, etc. A WSN is similar to mobile ad hoc networks, except that the sensors have limited battery power, limited transmission rate, and much reduced other capabilities. The limited embedded battery life-time of each sensor node calls for minimizing power consumption in the sensors, which is crucial for successful and reliable network operation. Thus, power and energy efficiency is of paramount interest, and the optimal WSN design should consume the minimum amount of power needed to provide reliable communication [9].

With the micro-electro-mechanical system (MEMS) technology, the size and cost of a sensor node have been significantly reduced. On the other hand, energy efficiency has also been significantly enhanced by incorporating energy awareness in the design of application and network protocols [4, 11, 12, 13]. However, energy-efficient reliable data transmission is still as schallenge in energy constrained WSNs. The wireless network of sensor nodes is inherently exposed to various sources of unreliability and integrity problems such as unreliable communication channels, transmission errors, node failures, malicious tampering of nodes and eavesdropping. In WSN, multi-user interference caused by densely populated sensors may lead to a high packet error rate. While retransmissions can be used to recover from data loss, it requires additional energy and introduces delay in trans-to-receive mode changeover. Sensor devices being energy constrained, retransmission based schemes have proved too costly. Therefore, one of the goals of the wireless sensor networks is to enable energy-efficient reliable data transmission and reception to meet the applications requirement [1]. The existing MANET routing approaches may not work optimally, and have prompted various new research studies [5-8].

It is well known fact that transmission on wireless channels is much more error prone than on wired channels. Physical phenomena like reflection, diffraction, and scattering of waveforms, partially in conjunction with moving nodes or movements in the environment, lead to fast fading and inter-symbol interference. Path loss, attenuation, and the presence of obstacles lead to slow fading. In addition, there is noise and interference from other nodes/other systems working in overlapping or neighboring frequency bands. Thus, transmission errors are inherent in wireless communications because of these instability of wireless channels, which is due to many reasons, for example, channel fading, time-frequency coherence, and inter-band interference [14]. The distortion of waveforms translates into bit errors and packet losses. A transmission packet having errors is useless and results into extra processing with respect to retransmission at both the sender and the receiver stations. The most important error control techniques data communications are Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC). ARQ, also known as Automatic Repeat Query, is an error-control method for data transmission that uses acknowledgements (messages sent by the receiver indicating that it has correctly received a data frame or packet) and timeouts (specified periods of time allowed to elapse before an acknowledgment is to be received) to achieve reliable data transmission over an unreliable service. If the sender does not receive an acknowledgment before the timeout, it usually re-transmits the frame/packet until the sender receives an acknowledgment or exceeds a predefined number of re-transmissions. In ARQ, damaged packets are retransmitted without trying to correct the error. There are three types of ARQ protocols namely Stop-and-wait ARQ, Go-Back-N ARQ, and Selective Repeat ARQ. These protocols reside in the Data Link or Transport Layers of the OSI model. The Transmission Control Protocol uses a variant of Go-Back-N ARQ to ensure reliable transmission of data over the Internet Protocol, which does not provide guaranteed delivery of packets; with Selective Acknowledgement (SACK), it uses Selective Repeat ARQ. ARQ has advantages that it produces no overheads and has optimal throughput in a non-error situation. However, the main disadvantages of ARQ are the huge delays and retransmission costs when an error occurs and results into huge energy loss.

Forward error correction (FEC) (also called channel coding) is a system of error control for data transmission, whereby the sender adds systematically generated redundant data to its messages, also known as an error-correcting code. The carefully designed redundancy allows the receiver to detect and correct a limited number of errors occurring anywhere in the message without the need to ask the sender for additional data. FEC gives the receiver an ability to correct errors without needing a reverse channel to request retransmission of data, but this advantage is at the cost of a fixed higher forward channel bandwidth. FEC is therefore applied in situations where retransmissions are relatively costly, or impossible such as when broadcasting to multiple receivers. In particular, FEC information is usually added to mass storage devices to enable recovery of corrupted data.

Thus, when designing wireless networks, the link reliability and energy-efficiency are important parameters for sensor networks, due to the nature of channels in various application scenarios [15]. Though the error control codes have higher power efficiency these conventional approaches like FEC or ARQ all require acknowledgement. Acknowledgement is needed in ARQ to let the sender know whether it needs to retransmit the packet. FEC eliminates the need for retransmission, but acknowledgement is still needed to let the sender know if the packet has been received at all by the destination, no matter it is correct or not. Acknowledgement is very expensive in terms of power [16]. The problem with the current protocols is that they find lowest energy route on the network, and use it always for every Communication, though it is not the best method to increase the network lifetime. Using the same low energy path also leads to network partition [18].

In this paper, we propose energy efficient transmission error recovery for WSNs. The paper is organized as follows. Section 2 summarizes the related work. Section 3 describes the proposed transmission error recovery algorithms with the help of illustrative examples. Simulation results are presented in Section 4. Conclusion is given in Section 5.

2. Related Work

There has been several research papers published on error control techniques in wireless sensor networks. Though these research publications have tried to focus on efficient routing schemes to optimize on the power usage of the WSN, none of them are directly applicable to the energy-efficiency aspect of the WSN in case of error encountered particularly during transmission and reception. Also, each technique has advantages and disadvantages. Battery power of individual sensor nodes is a precious resource in the WSN [2, 3]. For example, the power consumed by a Berkeley mote to transmit 1-bit of data is equivalent to the computation of 800m instructions [2]. When the battery power at a sensor node expires, the sensor node discontinues its operations in the network. Therefore, preserving the battery power of the individual sensor nodes is one of the primary concerns that pervades the design and operations of the WSN. The larger life span for battery power of individual sensor nodes may be achieved by minimizing the error in transmission and optimized re-transmission of messages/packets.

Havinga has studied energy-efficiency of FECs over various retransmission schemes for wireless communications [19]. As compared in [20, 21], fixed and small error correction codes are not effective in reducing packet losses when the errors are busty. In [22], an adaptive FEC scheme based on Universal Reed Solomon decoder has been presented for reliability in WSN. This scheme is particularly suitable for applications requiring high packet reliability and in which transmitting nodes consist of diversified features in terms of sensory data and/or coding schemes. However, such adaptive universal FEC has drawback of non-zero overheads in no-error and low-error situations compared to ARQ protocol. Some times multi-path routing algorithms are used for ensuring error-free data packet reception at the destination in a sensor network. However, this method has the disadvantage of increasing the overall traffic and energy usage substantially. In [23], the original data packet is split in sub-packets and each one of them is sent through one of the multiple paths. The sub-packets are created using erasure or FEC codes that add redundancy to the original source data. The total number of sub-packets as well as the added redundancy is a function dependent on the multipath degree and on the failing probabilities of the available paths. In this case, sequence of sub-packets is transmitted out of which only few sub-packets are necessary to reconstruct the original packet instead of the whole packet. Thus, the original message is constructed even if some of them are lost. But estimating the failing probabilities of each node is a challenging task. In [24], a meshed multipath routing (M-MPR) protocol with selective forwarding of packets and end-to-end forward error correction coding has been presented. It allows some intermediate nodes to have more than one forwarding direction to a given destination. In addition, selective forwarding of packets is done where the forwarding decision is taken dynamically, hop-by-hop, based on the conditions of downstream forwarding nodes. End-to-end FEC coding is also used to avoid acknowledgement-based retransmission.

In [25], a technique has been introduced which uses the temporal correlation of the data to correct transient errors in the received data using a data prediction model and a-posteriori information about future data. The suggested correction technique, however, assumes a perfect knowledge of the data properties and uses a pre-characterized data model to assist the correction process. However, building a perfect model of data offline is practically infeasible at times because of the following reasons. In a real sensor network, the data properties are context dependent. For example, temperature variation in a sensor being deployed outside in the field is different than one deployed inside an apartment. Moreover, the data properties vary over the lifetime of the sensor application necessitating run time changes in the data model. In [1], the earlier framework has been enhanced to model based error correction framework by introducing a scalable data modeling methodology to correct transient errors in the data received from the sensor nodes.

3. Proposed Energy Conservation Algorithm Scheme

As we know that basic steps of data communications are packet formation, transmission, reception, and packet interpretation. Figure 1 illustrates the basic operation of data communication.

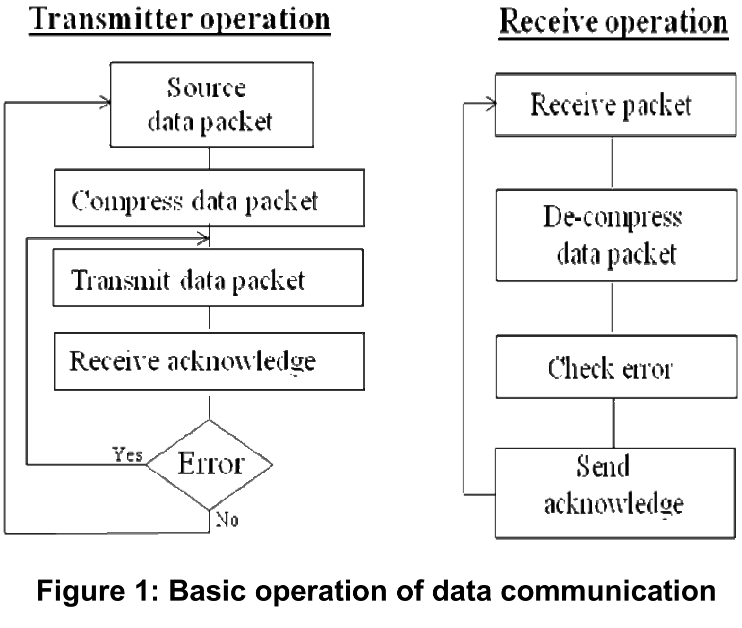

Figure 1: Basic operation of data communication

Recovery of transmitted data packet, in case, of error, can be achieved through different methods in data communications as mentions below.

Method 1: Resending the entire packet if corrupt.
Method 2: Resending only the corrupted portion of the packet.
Method 3: Breaking the packets into small sub-packets and resending only the corrupted sub-packet.
Method 4: Breaking the packets into small sub-packets and re-sending the only the corrupted portion of the sub-packet.

The emphasis of our approach is to increase the life span of the network by retransmission of only corrupted portion of the sub-packet and not the whole sub-packet. The activities performed at transmitting and receiving sides are illustrated below in tabular form for easy understanding for each of the methods.

Method 1: Re-sending the entire packet if corrupt.

Transmitter	Receiver
Compress data packet before transmitting	
Transmit the entire packet	⇨ Receive packet and de-compress
	Check error (only OK / Not OK)
Receive acknowledgement	⇦ Send acknowledgement (error found)
Retransmit the entire packet if faulty	⇨ Receive packet and de-compress
	Check error (only OK / Not OK)
Receive acknowledgement	⇦ Send acknowledgement (no error found)

Method 2: Re-sending only the corrupted portion of the packet.

Transmitter	Receiver
Compress data packet before transmitting	
Transmit the entire packet	⇨ Receive packet and de-compress
	Check error (algorithm to check the % of data packet received OK)
Receive acknowledgement	⇦ Send acknowledgement (% of OK packet received)
Create new packet of the part of data not received by the Receiver from the original packet	
Compress data packet before transmitting	
Retransmit the new sub-packet	⇨ Receive new sub-packet and de-compress
	Check error (algorithm to check the % of data packet received OK)
Receive acknowledgement	⇦ Send acknowledgement (no error found)

Method 3: Breaking the packets into small sub-packets and re-sending only the corrupted sub-packet.

Transmitter	Receiver
Create Sub-packets of the entire packet	
Compress data sub-packets before transmitting	
Transmit the sub-packet	⇨ Receive sub-packet and de-compress
	Check error (only OK / Not OK)
Receive acknowledgement	⇦ Send acknowledgement (error found)
Retransmit faulty sub-packet	⇨ Receive sub-packet and de-compress
	Check error (only OK / Not OK)
Receive acknowledgement	⇦ Send acknowledgement (no error found)
Repeat the process till all the packets are sent successfully	

Method 4: Breaking the packets into small sub-packets and re-sending only the corrupted portion of the sub-packet.

Transmitter	Receiver
Create Sub-packets of the entire packet	
Compress data sub-packet before transmitting	
Transmit the sub-packet	⇨ Receive sub-packet and de-compress
	Check error (algorithm to check the % of data sub-sub-packet received OK)
Receive acknowledgement	⇦ Send acknowledgement (% of OK sub-sub-packet received)
Create new sub-sub-packet of the part of data not received by the Receiver from the original sub-packet	
Compress data sub-sub-packet before transmitting	
Retransmit the new sub-sub-packet	⇨ Receive sub-sub-packet and de-compress
	Check error (algorithm to check the % of data sub-sub-packet received OK)
Receive acknowledgement	⇦ Send acknowledgement (% of OK sub-sub-packet received)
Repeat the process till all the sub-packets are sent successfully	

3.1 Basic Assumptions of Parameters

- Assuming distance between the Transmitter & Receiver is 50meters
- Time required for transmitting 1bits of data is 6nSec [17]
- Battery Voltage is 9Volts
- Average current required for transmitting 1bit of data is 24mAmps

- Average current required in idle state (transmitter/receiver) is 500µAmps
- Average current required for processing (data compression, sub-packet creation, etc.) 1bit of data is 750µAmps
- Data Packet is 100 bits, Acknowledge Packet is 5 bits
- Error occurs after 30% of the packet is received by the receiver
- Length of Sub-Packets is 20bits

The basic formulas for energy calculations is given as

Energy (E) = Power (P) * Time (T)
Power (P) = Voltage (V) * Current (I)

Now based on the above formula, the per bit energy calculation is given in Table 1.

Table 1: Per bit energy calculation

SN	Per Bit	Voltage	current(A)	Power(W)	Time(nSec)	Energy(nJoules) / bit
1	Transmission	9	0.024	0.216	6	1.296
2	Processing	9	0.00075	0.00675	6	0.0405
3	Receiving	9	0.018	0.162	6	0.972
4	Idle State	9	0.0005	0.0045	6	0.027

On the basis of above energy calculation, the power assumptions for transmitter and receiver are given in Table 2 and Table 3 respectively.

Table 2: Power assumptions for the various processes of the transmitter

1	Compression of data packet before transmitting (per bit)	$E_{compress}$	0.04nJ
2	Transmission of data packet (per bit)	$E_{Transmit}$	1.29 nJ
3	Receiving of data packet at transmitter(per bit)	$E_{Receive}$	0.97nJ
4	Creation of data sub-packet from the complete packet (per bit)	E_{sub_packet}	0.04nJ
5	Size of the data packet (in bits)	B_{Data_packet}	100 Bits
6	Size of error packet (in bits)	B_{error_packet}	40 Bits
7	Size of error sub-packet (in bits)	$B_{error\_sub\_packet}$	14 Bits
8	Size of sub packet (in bits)	B_{sub_packet}	20 Bits

Table 3: Power assumptions for the various processes of the receiver

1	Receiving data packet = Receiving of data packet at transmitter (per bit)	$E_{Receive}$	0.97nJ
2	De-compression of data packet (per bit) = Compression of data packet before transmitting (per bit)	$E_{compress}$	0.04nJ
3	Error checking (per bit)	E_{error_check}	0.04nJ
4	Transmit acknowledge = Transmission of data packet (per bit)	$E_{Transmit}$	1.29 nJ
5	Error checking - Algorithm to check the % of data packet received OK (per bit)	$E_{error\_Algorithm}$	0.04nJ
6	Size of error check packet (in bits)	B_{error_check}	100 Bits
7	Size of acknowledgement packet (in bits)	$B_{Acknowledge}$	5 Bits

Now the energy computations for different methods are given below.

Method 1: Resending the entire packet if corrupt.

$E_{Transmitter} = E_{compress} * B_{Data\_packet} + 2*(E_{Transmit} * B_{Data\_packet} + E_{Receive} * B_{Acknowledge})$
$= 0.04 * 100 + 2*(1.296*100 + 0.972*5)$
$= 272.92$ nJ

$E_{Receiver} = 2*(E_{Receive} * B_{Data\_packet} + E_{compress} * B_{Data\_packet} + E_{error\_check} * B_{error\_check} + E_{Transmit} * B_{Acknowledge})$
$= 2*(0.972*100 + 0.04*100 + 0.04*100 + 1.296*5)$
$= 223.36$ nJ

$E_{total\_method1} = E_{Transmitter} + E_{Receiver}$
$= 496.28$ nJ

Method 2: Resending only the corrupted portion of the packet.

$E_{Transmitter} = E_{compress} * B_{Data\_packet} + E_{Transmit} * B_{Data\_packet} + 2*(E_{Receive} * B_{Acknowledge}) + E_{sub\_packet} * B_{error\_packet} + E_{compress} * B_{error\_packet} + E_{Transmit} * B_{error\_packet})$
$= 0.04*100 + 1.296*100 + 2*(0.972*5) + 0.04*40 + 0.04*40 + 1.296*40$
$= 239.64$ nJ

$E_{Receiver} = E_{Receive} * B_{Data\_packet} + E_{compress} * B_{Data\_packet} + E_{error\_algoritm} * B_{Data\_packet} + 2*(E_{Transmit} * B_{Acknowledge}) + E_{Receive} * B_{error\_packet} + E_{compress} * B_{error\_packet} + E_{error\_algoritm} * B_{error\_packet}$
$= 0.972*100 + 0.04*100 + 0.04*100 + 2*(1.296*5) + 0.972*40 + 0.04*40 + 0.04*40$
$= 191.8$ nJ

$E_{total\_method2} = E_{Transmitter} + E_{Receiver}$
$= 431.44$ nJ

Method 3: Breaking the packets into small sub-packets and re-sending only the corrupted sub-packet.

$E_{Transmitter} =$ No. of Sub-packets$*(E_{sub_packet} * B_{sub_packet} + E_{compress} * B_{sub_packet}) +$ (No. of Sub-packets$+1)*(E_{Transmit} * B_{sub_packet} + E_{Receive} * B_{Acknowledge})$
$= 5*(0.04*20 + 0.04*20) + 6*(1.296*20 + 0.972*5)$
$= 192.68$ nJ

$E_{Receiver} =$ (No. of Sub-packets$+1)*(E_{Receive} * B_{sub_packet} + E_{compress} * B_{sub_packet} + E_{error_check} * B_{sub_packet} + E_{Transmit} * B_{Acknowledge})$
$= 6*(0.972*20 + 0.04*20 + 0.04*20 + 1.296*5)$
$= 165.12$ nJ

$$E_{total\_method3} = E_{Transmitter} + E_{Receiver}$$
$$= 357.8 \text{ nJ}$$

Method 4: Breaking the packets into small sub-packets and re-sending the only the corrupted portion of the sub-packet.

$$E_{Transmitter} = \text{No. of Sub-packets}*(E_{sub\_packet} * B_{sub\_packet} + E_{compress} * B_{sub\_packet} + E_{Transmit} * B_{sub\_packet} + E_{Receive} * B_{Acknowledge}) + E_{sub\_packet} * B_{error\_sub\_packet} + E_{Transmit} * B_{error\_Sub\_packet} + E_{Receive} * B_{Acknowledge}$$
$$= 5*(0.04*20 + 0.04*20 + 1.296*20 + 0.972*5) + 0.04*14 + 1.296*14 + 0.972*5$$
$$= 185.46 \text{ nJ}$$

$$E_{Receiver} = (\text{No. of Sub-packets})*(E_{Receive} * B_{sub\_packet} + E_{compress} * B_{sub\_packet} + E_{error\_algorithm} * B_{sub\_packet} + E_{Transmit} * B_{Acknowledge}) + E_{Receive} * B_{error\_sub\_packet} + E_{compress} * B_{error\_sub\_packet} + E_{error\_algoritm} * B_{error\_sub\_packet} + E_{Transmit} * B_{Acknowledge}$$
$$= 5*(0.972*20 + 0.04*20 + 0.04*20 + 1.296*5) + 0.972*14 + 0.04*14 + 0.04*14 + 1.296*5$$
$$= 158.8 \text{ nJ}$$

$$E_{total\_method4} = E_{Transmitter} + E_{Receiver}$$
$$= 344.27 \text{ nJ}$$

Figure 2 illustrates energy consumptions for different methods during retransmission due to transmission error occurrence assuming that the error was detected after successful transmission of 30% of data packet.

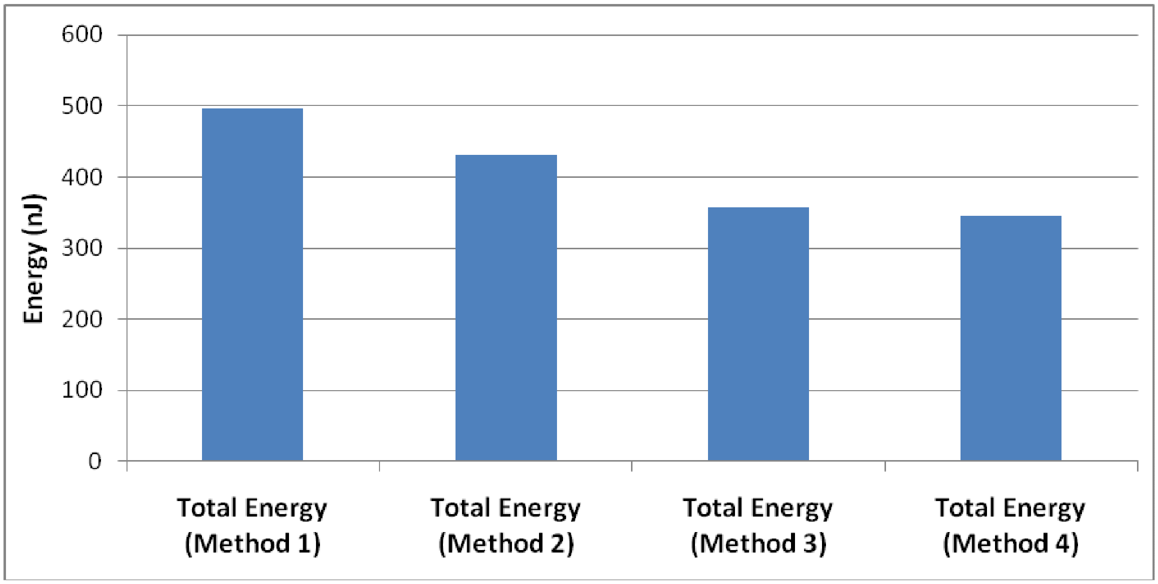

Figure 2: Energy computations for different methods during error occurrence

4. Simulation Results

Authentication of energy computation for the different methods of retransmission of corrupted packets can be performed in the following conditions:

Case 1: By keeping data packet and sub-packet size fixed with varying error occurrence percentage.
Case 2: By keeping data packet and error occurrence percentage fixed with varying sub-packet size.
Case 3: By keeping data packet fixed with varying sub-packet size and and error occurrence percentage.

Case 1: By keeping data packet and sub-packet size fixed with varying error occurrence percentage

- As shown in the Table 4 below, the percentage of error was varied from 0% to 100%
- The number of retries to sent successfully the corrupted sub-packet is 5

Table 4: Energy calculation for case 1

% of Packet delivered	Total Energy (Method 1)	Total Energy (Method 2)	Total Energy (Method 3)	Total Energy (Method 4)
0	14196	14596	3298	3308
10	14196	13382	3298	3248
20	14196	12168	3298	3189
30	14196	10954	3298	3129
40	14196	9740	3298	3069
50	14196	8526	3298	3010
60	14196	7312	3298	2950
70	14196	6098	3298	2890
80	14196	4884	3298	2830
90	14196	3670	3298	2771
100	14196	2456	3298	2711

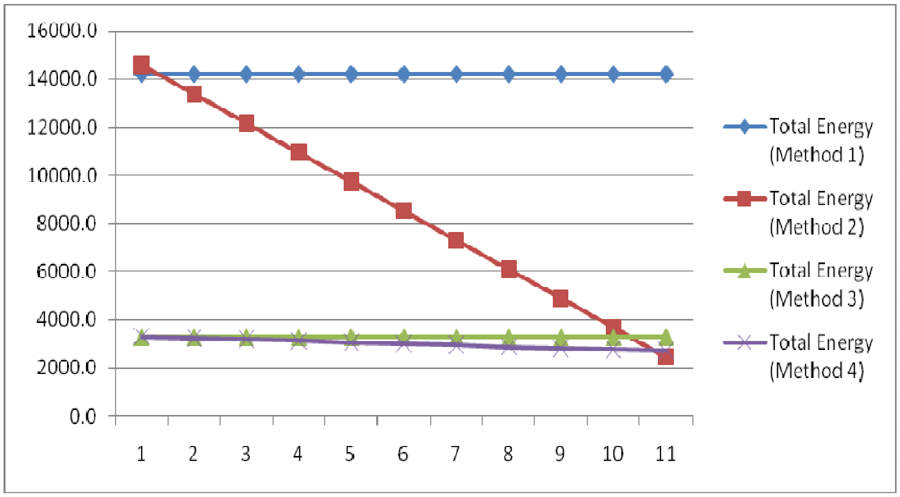

Figure 3: Enegry plot for case 1

Conclusion for Case 1: After comparing the minimum and maximum energy usage for the entire range of the experiment, it is evident from the Figure 3 that **Method 4** is most optimised.

Case 2: By keeping data packet and error occurrence percentage fixed with varying sub-packet size:

- As shown in Table 5 below, the percentage of error fixed at 30%
- The number of retries to sent successfully the corrupted sub-packet is 5
- The data packet is of 1000 bits
- The sub-packet size was varied from 5 bits to 500 bits

Table 5: Energy calculation for case 2

% of Packet delivered	Total Energy (Method 1)	Total Energy (Method 2)	Total Energy (Method 3)	Total Energy (Method 4)
5	14196	10954	4811	4794
10	14196	10954	3736	3702
20	14196	10954	3286	3218
25	14196	10954	3231	3147
50	14196	10954	3298	3129
100	14196	10954	3772	3433
125	14196	10954	4042	3620
200	14196	10954	4889	4213
250	14196	10954	5465	4619
500	14196	10954	8377	6686

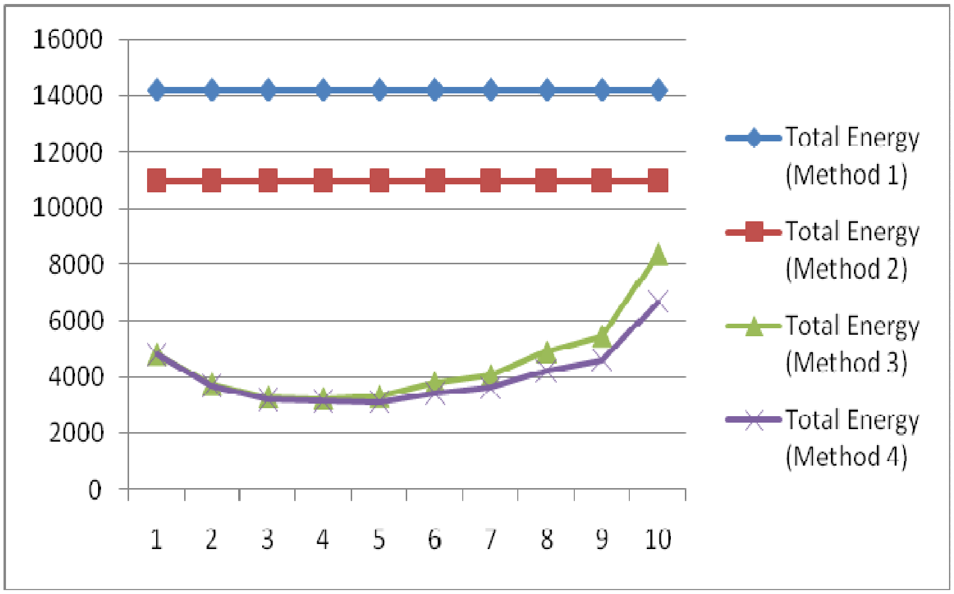

Figure 4: Enegry plot for case 2

Conclusion for Case 2: After comparing the minimum and maximum energy usage for the entire range of the experiment, it is evident from the Figure 4 that **Method 4** is most optimised and the best sub-packet size in this case is 50 bits

Case 3: By keeping data packet fixed with varying sub-packet size and error occurrence percentage:

- As shown in Table 6 below, the number of retries to sent successfully the corrupted sub-packets is 5
    - The data packet is of 1000 bits
    - The sub-packet size was varied from 5 bits to 500 bits

Table 6: Energy calculation for case 3

Sub-packet size (bits)	25% error Method 1	25% error Method 2	25% error Method 3	25% error Method 4	50% error Method 1	50% error Method 2	50% error Method 3	50% error Method 4	75% error Method 1	75% error Method 2	75% error Method 3	75% error Method 4
5	14196	11561	4811	4797	14196	8526	4811	4783	14196	5491	4811	4768
10	14196	11561	3736	3708	14196	8526	3736	3678	14196	5491	3736	3649
20	14196	11561	3287	3231	14196	8526	3287	3171	14196	5491	3287	3111
25	14196	11561	3232	3162	14196	8526	3232	3088	14196	5491	3232	3013
50	14196	11561	3299	3159	14196	8526	3299	3010	14196	5491	3299	2861
100	14196	11561	3772	3494	14196	8526	3772	3195	14196	5491	3772	2897
125	14196	11561	4043	3695	14196	8526	4043	3322	14196	5491	4043	2949
200	14196	11561	4889	4332	14196	8526	4889	3735	14196	5491	4889	3138
250	14196	11561	5465	4769	14196	8526	5465	4023	14196	5491	5465	3276
500	14196	11561	8377	6985	14196	8526	8377	5492	14196	5491	8377	4000

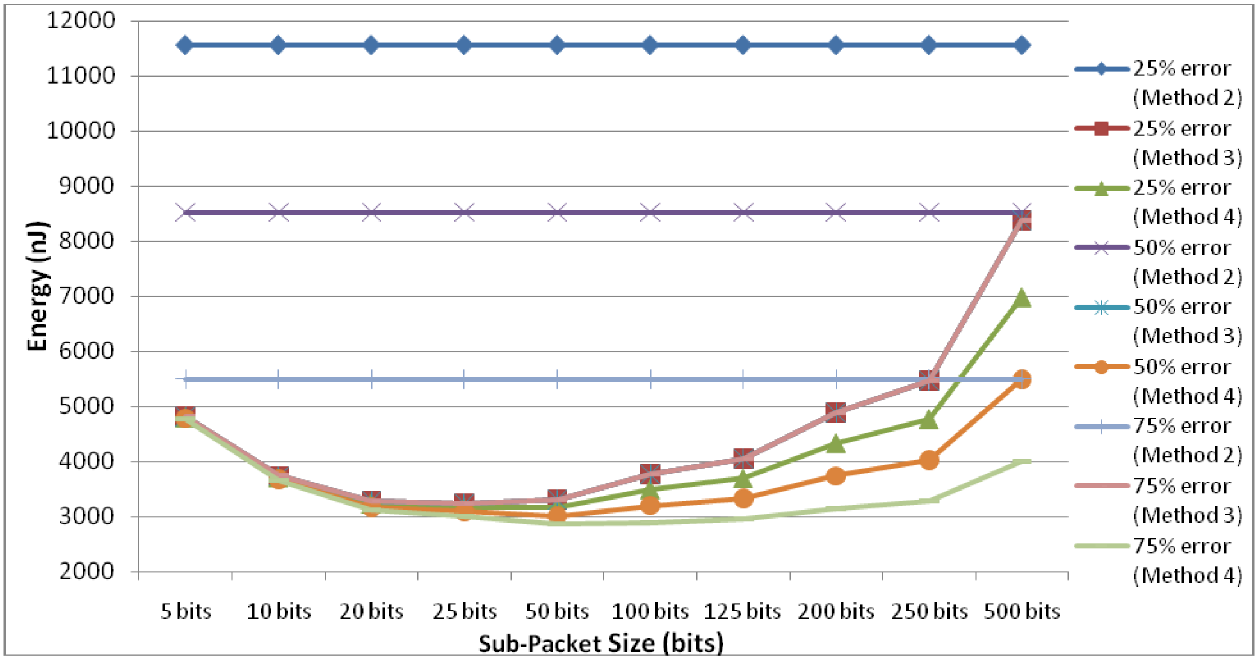

Figure 5: Enegry Plot for Case 3

Conclusion for Case 3: After comparing the minimum and maximum energy usage for the entire range of the experiment, it is evident from the Figure 5 that **Method 4** is most optimised and the greater the Percentage of error the better is the performance.

5. Conclusion

In this paper we have illustrated an energy-efficient transmission error recovery algorithm for wireless sensor network that saves power and prolongs life of the sensor network. The algorithm helps in deciding the optimum size of data sub-packet. The life span of the network is increased by breaking of the message packet into small number of sub-packets and resending not the total sub-packets but only the corrupted portion of the sub-packets. With various permutations and combinations, the simulation results show that by breaking the packets into small sub-packets and resending only corrupted portion of the sub-packet is the most efficient way of data transmission and error recovery in wireless networks, as shown in Figure 5. The simulation results show that more than 30% of battery power savings can be achieved by implementing proposed system with respect to the conventional retransmission of complete packet.

References

[1.] S. Mukhopadhyay, D. Panigrahi, and S. Dey, “Model Based Error Correction for Wireless Sensor Networks”.

[2.] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Tag Hong, “A Tiny Aggregation Service for Ad-Hoc Sensor Networks” *SIGOPS Opemting System Review* 96, SI (December 2002), pp. 131-146.

[3.] G.J. Pottle, and W.J. Kaiser, “Embedding the Internet: Wireless Integrated Network Sensors”, *Communications of the ACM* 43,5 [May 2000], pp. 51-58.

[4.] S.K. Singh, M.P. Singh, and D.K. Singh, "Energy-efficient Homogeneous Clustering Algorithm for Wireless Sensor Network", *International Journal of Wireless & Mobile Networks (IJWMN)*, Aug. 2010, vol. 2, no. 3, pp. 49-61.

[5.] S.B. Qaisar and H. Radha, "Optimal Progressive error Recovery for Wireless Sensor Networks using Irregular LDPC Codes", CISS Paper#80.

[6.] M. Sartipi, F. Fekri, "Source and Channel Coding in Wireless Sensor Networks using LDPC Codes,'' *ICSACN*, pp. 309-316, October 2004.

[7.] E. shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," *ACM SIGMOBILE*, July 2001.

[8.] Y. Sankarasubramaniam, I. F. Akyildiz, and S. W. McLaughlin, "Energy efficiency based packet size optimization in wireless sensor networks," *IWSN*, 2003.

[9.] S.L. Howard, C. Schlegel, and K. Iniewski, "Error Control Coding in Low-Power Wireless Sensor Networks: When Is ECC Energy-Efficient?", *EURASIP Journal on Wireless Communications and Networking*, March 2006, pp. 1-14.

[10.] S.K. Singh, M.P. Singh, and D.K. Singh, "Applications, Classifications, and Selections of Routing Protocols for Wireless Sensor Networks" *International Journal of Computer Networks and Communications (IJCNC)*, Nov. 2010, vol. 2, no. 3.

[11.] S.K. Singh, M.P. Singh, and D.K. Singh, "Routing Protocols in Wireless Sensor Networks – A Survey" *International Journal of Computer Science and Engineering Survey (IJCSES)*.

[12.] S.K. Singh, M.P. Singh, and D.K. Singh, "A survey of Energy-Efficient Hierarchical Cluster-based Routing in Wireless Sensor Networks", *International Journal of Advanced Networking and Application (IJANA)*, Sept.-Oct., 2010, vol. 02, issue 02, pp. 570-580.

[13.] S.K. Singh, M.P. Singh, and D.K. Singh, "Performance Evaluation and Comparison of Energy-efficient Routing Protocols for Wireless Sensor Network", *International Journal of Wireless & Mobile Networks (IJWMN)*

[14.] Jun Zheng and Abbas Jamalipour, "Wireless Sensor Networks: A Networking Perspective", a book published by A John & Sons, Inc, and IEEEE, 2009.

[15.] F. Ian Akyildiz and Y.S. Su Weilian, "A Survey on Sensor Networks", *IEEE Communications Magazine* Georgia Institute of Technology, 2002.

[16.] Lizhi Charlie Zhong, Rahul Shah, C. G. An Ultra-Low Power and Distributed Access Protocol for Broadband Wireless Sensor Networks. *IEEE Broadband Wireless Summit, Las Vegas, N.V.* 2001.

[17.] J.L. da Silva Jr. and M. J. Shamberger, "Design Methodology for PicoRadio Networks 2001".

[18.] R. C. Shah and J. M. R., "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks", *IEEE Wireless Communications and Networking Conf.,* Orlando, FL 2002.

[19.] P. J. M. Havinga, "Energy Efficiency of Error Correcting Mechanisms for Wireless Communications", *Proc., IEEE Wireless Communications and Networking Conf.*, 1999, pp. 616–620.

[20.] S. Kim, R. Fonseca and D. Culler "Reliable Transfer on Wireless Sensor Networks", *Proc., IEEE SECON*, 2004, pp. 449–459.

[21.] J. Jeong and C. T. Ee "Forward Error Correction in Sensor Networks", *Technical Reports, University of California Berkeley*, 2003.

[22.] R. Agarwal, E.M. Popovici and B. O'Flynn, "Adaptive Wireless sensor Networks: A System Design Approach to Adaptive Reliability".

[23.] S. Dulman, Tim Nieberg, Jian Wu and Paul Havinga, "Trade-off between Traffic Overhead and Reliability in Multipath Routing for Wireless Sensor Networks".

[24.] S. De, C. Qiao and H. Wu, "Meshed multipath routing with selective forwarding; an efficient strategy in wireless sensor networks", *presented in WCNC"* (Elsevier Computer Networks), 2003, pp. 481-497.

[25.] S. Bandopadhyay, D. Panigrahi, and S. Dey, "Data aware, low cost error correction for wireless sensor networks", in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, March 2004, pp. 2492-7.

Authors

Shio Kumar Singh is Head of Maintenance Engineering Department (Electrical) at Tata Steel Limited, Jamshedpur, India. He received degrees in both Electrical and Electronics engineering, as well as M.Sc.(Engg.) in Power Electronics from Regional Institute of Technology, Jamshedpur, India. He also obtained "Executive Post Graduate Diploma in International Business" from Indian Institute of Foreign Trade (IIFT), New Delhi, India. He is an accomplished academician with rich industrial experience in design, development, implementation and marketing & sales of IT, Automation, and Telecommunication solutions, Electrical & Electronics maintenance, process improvement initiatives (Six-sigma, TPM, TOC), and Training & Development in a distinguished career spanning over 30 years. He has published number of papers in both national and international journals and has presented these in various seminars and symposiums.

He is author of several engineering books such as Database Management System, Industrial Instrumentation and Control, and Process Control Systems published by Pearson Education, McGraw-Hill, and Prentice-Hall of India. He is widely traveled and has visited various industries in Europe and South Asian countries for study and marketing of process automation systems. He has been conferred the Eminent Engineer and Distinguished Engineer Awards by The Institution of Engineers (India) for his contributions to the field of computer science and engineering. He is a Chartered Engineers and also a Fellow Member (FIE) of The Institution of Engineers (India).

Dr. M. P. Singh is an Assistant Professor in the Department of Computer Science and Engineering at National Institute of Technology Patna, Bihar, India. He has experience of five years. He has authored number of papers which have been published in both national and international journals. His research interest is in the area of Wireless Sensor Network, Mobile Computing

Dr. Dharmendra K Singh is presently working as Head, Department of Electronics and Communication & Information Technology, BIT Sindri, Dhanbad. He has more than 20 years of teaching experience. He is heading the department of Electronics and Communication & Information technology since 2002. He is instrumental in starting the curriculum on information technology. He has published more than 35 papers in journals and conferences. He has already supervised 01 thesis in computer Science & Engg and 05 research scholars are presently enrolled for their doctoral degree. The area of research he works are Coding theory, cryptography, optical Amplifiers, Photonic Crystal Fibers, e-Governance and Educational Planning. He is member and conveners of various computerization programs of BIT Sindri, Vinoba Bhave University, Ranchi University. He is also a Fellow Member (FIE) of The Institution of Engineers (India).