International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 14 (2017) pp. 4468-4473

Multiple Forward Error Correction in ARQ Systems by Combining Three Consecutive Erroneous Frames in a Wireless Sensor Network

Shylashree N[1] and Pratibha K[2]

[1]*Associate Professor VTU, Department of ECE, RV College of Engineering, Bengaluru-59, Karnataka, India.*

Orcid: 0000-0003-4185-6190

[2] *Assistant Professor, VTU, Department of ECE, RV College of Engineering, Bengaluru-59, Karnataka, India.*

Orcid: 0000-0002-7637-6675

Abstract

A new technique of forward error correction in an automatic repeat request system in a wireless sensor network is described. At the receiver, we make use of pair-wise xor combining among three consecutive erroneous frames to detect the locations of multiple errors and then correct them.

Keywords: ARQ, Multiple Forward error correction, HARQ, Packet Success probability.

INTRODUCTION

Consider a multi-hop Wireless Sensor Network (WSN), where the sensor nodes send their data to the sink (or Base station). In a conventional Automatic Repeat Request (ARQ) system, if a data frame is received with one or more bit errors, the same frame is retransmitted from the sensor nodes to the sink until the requisite acknowledgement arrives from the receiver. This repetition of transmissions causes additional depletion of battery energy in sensors. We can save this extra loss of energy using Forward Error Correction at the receiver by reducing the frame retransmission count. Thereby, the working life of sensor nodes is prolonged. At present, we have several FEC coding methods like Reed-Solomon, Low-Density-Parity-Check (LDPC), Golay, BCH, Turbo *etc*. These codes have higher computationally complexity and demand more processing resources. Therefore, in this paper, a simple FEC scheme with relatively low computational overhead is presented. This scheme belongs to Hybrid Automatic Repeat Request (HARQ) system [1], [2], [3] with notable modifications. In a HARQ system, the frame received with errors is not discarded. It is stored and logically combined with the immediately next repetition of the same frame to correct the errors. In our scheme, we use a standard error detection scheme like CRC for each frame. Here, we consider the case of multiple bit errors in received frames for error correction. In our scheme, three consecutive incorrectly received frames derived from the same source frame are logically combined to locate the position of the error. We use Chase Combining [4], [5] in our method.

NOISEMODEL

The assumed noise model is shown in Fig.1. Here the source is a sensor node and the destination is the sink. Let the transmitted data be represented by a binary vector **A** of size N bits as,

$$\mathbf{A} = [\ a(1)\ \ a(2)\ \ \ldots.\ \ a(N)\ ] \tag{1}$$

We assume that a standard Error Detecting CRC is included in the transmitted frame. The error is modelled by the random binary vector **E** of size N, as shown in Fig.1. The elements of **E** are represented as,

$$\mathbf{E} = [\ e(1)\ \ e(2)\ \ \ldots.\ \ e(N)\ ] \tag{2}$$

The received vector at the destination is given by **B** as,

$$\mathbf{B} = \mathbf{A} \oplus \mathbf{E} \tag{3}$$

Here the bitwise **xor** is represented by the symbol $\oplus$. The size of **B** is N and its elements are given by,

$$\mathbf{B} = [\ b(1)\ \ b(2)\ \ \ldots\ \ b(N)] \tag{4}$$

Because of relation (3),

$$b(i) = a(i) \oplus e(i) \tag{5}$$

for i = 1, 2,…, N.

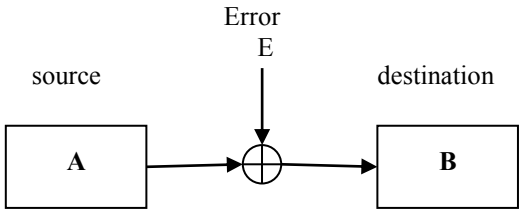

Figure 1. Noise Model

From (5), we see that the number of 1's in **E** gives the number of received error bits. The locations of 1's in **E** give the locations of errors in **B.** The number of errors can be one or more. Our objective is to determine the locations of the error

and to correct them. The 0's of **E**, correspond to the error-free bits of **B**.

BASIC PRINCIPLE AND WORKING

Let the present transmitted data vector under consideration be **A**. Let the corresponding received data vector be denoted by $\mathbf{B_1}$. Let there be some bit errors in $\mathbf{B_1}$. It is detected by the receiver and an ARQ NAK frame is sent by the receiver to the source. We assume that the NAK frame is effectively error free. The source in turn retransmits **A** and the corresponding received frame is denoted by $\mathbf{B_2}$. If $\mathbf{B_2}$ is error- free, then there is no need for any correction and we can go to receive the next fresh frame. However, when $\mathbf{B_2}$ has errors along with $\mathbf{B_1}$, the sender retransmits **A** which is again received as $\mathbf{B_3}$. If $\mathbf{B_2}$ is error- free, then there is no need for any FEC and the requisite ACK is sent by the receiver. If $\mathbf{B_3}$ also has errors, then using our proposed method, we correct the errors using the three consecutive received data vectors $\mathbf{B_1}$, $\mathbf{B_2}$ and $\mathbf{B_3}$.

A. Basic assumptions

Let the error vectors be $\mathbf{E_1}$, $\mathbf{E_2}$ and $\mathbf{E_3}$ corresponding to the received vectors $\mathbf{B_1}$, $\mathbf{B_2}$ and $\mathbf{B_3}$. Then,

$$\mathbf{B_1} = \mathbf{A} \oplus \mathbf{E_1} \quad (6)$$

$$\mathbf{B_2} = \mathbf{A} \oplus \mathbf{E_2} \quad (7)$$

$$\mathbf{B_3} = \mathbf{A} \oplus \mathbf{E_3} \quad (8)$$

Here, $\mathbf{E_1}$, $\mathbf{E_2}$ and $\mathbf{E_3}$ are random binary vectors in the sense that the locations of 1's in them are random and unknown. Our objective is to determine the locations of 1's in $\mathbf{E_1}$, $\mathbf{E_2}$ and $\mathbf{E_3}$.

Let the number of 1's (errors) in $\mathbf{E_1}$, $\mathbf{E_2}$ and $\mathbf{E_3}$ be designated by m_1, m_2 and m_3 respectively. In general m_1, m_2 and m_3 are small compared to N. For a given set of $\mathbf{E_1}$, $\mathbf{E_2}$ and $\mathbf{E_3}$, the values of m_1, m_2 and m_3 are unknown. The errors in $\mathbf{E_1}$, $\mathbf{E_2}$ and $\mathbf{E_3}$ are independent and identically distributed (i.i.d) with uniform distribution. We make the following important assumptions about $\mathbf{E_1}$, $\mathbf{E_2}$ and $\mathbf{E_3}$.

Locations of error bits (1's) in $\mathbf{E_1}$, $\mathbf{E_2}$ and $\mathbf{E_3}$ are disjoint. This non-overlapping requirement is expressed algebraically as,

$$\mathbf{E_1E_2} = \mathbf{E_1.*E_2} = \text{bitand}(\mathbf{E_1},\mathbf{E_2}) = \mathbf{0} \quad (9)$$

$$\mathbf{E_1E_3} = \mathbf{E_1.*E_3} = \text{bitand}(\mathbf{E_1},\mathbf{E_3}) = \mathbf{0} \quad (10)$$

$$\mathbf{E_2E_3} = \mathbf{E_2.*E_3} = \text{bitand}(\mathbf{E_2},\mathbf{E_3}) = \mathbf{0} \quad (11)$$

Here, **0** is the all zero vector of size N. The term $\mathbf{E_1E_2}$ represents the Boolean **and** of $\mathbf{E_1}$ and $\mathbf{E_2}$. The term $\mathbf{E1.*E_2}$ represents element-wise arithmetic product. For binary vectors, $\mathbf{E_1E_2}$ and $\mathbf{E1.*E_2}$ give the same result. Similarly notations apply for $\mathbf{E_1E_3}$, $\mathbf{E_1.*E_3}$, $\mathbf{E_2E_3}$ and $\mathbf{E_2.*E_3}$. Our algorithm works only when (9), (10) and (11) are satisfied.

Consider the number of error bits m_1, m_2 and m_3. They have to be greater than 0's. Otherwise there is no error. But they have to be much smaller compared to N, so that (9), (10) and (11) are satisfied. In practical scenarios, these conditions are satisfied.

B. Determination of E_1, E_2 and E_3

Received binary vectors $\mathbf{B_1}$, $\mathbf{B_2}$ and $\mathbf{B_3}$ are known and using them, we calculate three binary vectors $\mathbf{F_{12}}$, $\mathbf{F_{13}}$ and $\mathbf{F_{23}}$ as,

$$\mathbf{F_{12}} = \mathbf{B_1} \oplus \mathbf{B_2} \quad (12)$$

$$\mathbf{F_{13}} = \mathbf{B_1} \oplus \mathbf{B_3} \quad (13)$$

$$\mathbf{F_{23}} = \mathbf{B_2} \oplus \mathbf{B_3} \quad (14)$$

Substituting for $\mathbf{B_1}$ and $\mathbf{B_2}$ from (6) and (7) in (12), we get,

$$\mathbf{F_{12}} = \mathbf{E_1} \oplus \mathbf{E_2} \quad (15)$$

Similarly, from (6), (8) and (13),

$$\mathbf{F_{13}} = \mathbf{E_1} \oplus \mathbf{E_3} \quad (16)$$

Similarly, from (7), (8) and (14),

$$\mathbf{F_{23}} = \mathbf{E_2} \oplus \mathbf{E_3} \quad (17)$$

Now, consider the truth table of **XOR** operation as shown in Table 1. Here, the term $\mathbf{E_1E_2}$ represents Boolean **AND** as well as arithmetic multiplication depending on the context. Hence $\mathbf{2E_1E_2}$ of the last column in Table 1 represents arithmetic multiplication.

Table 1. Truth table for **xor** representation

E_1	E_2	$E_1 \oplus E_2$	E_1E_2	$E_1 + E_2$ Arithmetic	$E_1 + E_2 - 2E_1E_2$ Arithmetic
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	0	1	1
1	1	0	1	2	0

From Table 1, we see that $\mathbf{E_1} \oplus \mathbf{E_2}$ can be expanded as,

$$\mathbf{E_1} \oplus \mathbf{E_2} = \mathbf{E_1} + \mathbf{E_2} - \mathbf{2E_1E_2} \quad (18)$$

Here, $\mathbf{2E_1E_2}$ represents the scalar multiplication of vector $\mathbf{E_1E_2}$ by 2. From (18) and (15),

$$\mathbf{F_{12}} = \mathbf{E_1} + \mathbf{E_2} - \mathbf{2E_1E_2} \quad (19)$$

Similarly, (16) and (17) can be expanded as,

$$\mathbf{F_{13}} = \mathbf{E_1} + \mathbf{E_3} - \mathbf{2E_1E_3} \quad (20)$$

$$\mathbf{F_{23}} = \mathbf{E_2} + \mathbf{E_3} - \mathbf{2E_2E_3} \quad (21)$$

Now, applying the constraints given by (9), (10) and (11) to (19), (20) and (21) respectively, we get,

$$\mathbf{F_{12} = E_1 + E_2} \quad (22)$$

$$\mathbf{F_{13} = E_1 + E_3} \quad (23)$$

$$\mathbf{F_{23} = E_2 + E_3} \quad (24)$$

Now, consider the product$\mathbf{F_{12}F_{13}}$represented by $\mathbf{G_1}$ as,

$$\mathbf{G_1 = F_{12}F_{13}} \quad (25)$$

From (25), (22) and (23),

$$\mathbf{G_1 = (E_1 + E_2)(E_1 + E_3)}$$

On Expanding, we get,

$$\mathbf{G_1 = E_1E_1 + E_2E_1 + E_1E_3 + E_2E_3} \quad (26)$$

From (9), (10), (11), $\mathbf{E_2E_1 = E_1E_3 = E_2E_3 = 0}$ and since, $\mathbf{E_1}$ is a binary vector, $\mathbf{E_1E_1 = E_1}$.Therefore, (26) reduces as,

$$\mathbf{G_1 = E_1} \quad (27)$$

By, similar procedure, we can show that,

$$\mathbf{G_2 = F_{12}F_{23} = E_2} \quad (28)$$

$$\mathbf{G_3 = F_{13}F_{23} = E_3} \quad (29)$$

Therefore, the error vectors $\mathbf{E_1}$, $\mathbf{E_2}$ and $\mathbf{E_3}$ are determined as,

$$\mathbf{E_1 = G_1 = F_{12}F_{13}} \quad (30)$$

$$\mathbf{E_2 = G_2 = F_{12}F_{23}} \quad (31)$$

$$\mathbf{E_3 = G_3 = F_{13}F_{23}} \quad (32)$$

C. Recovery of the error free data vector A

Once, $\mathbf{E_1}$, $\mathbf{E_2}$ and $\mathbf{E_3}$ are known, A can be obtained using (6), (7) and (8) as,

$$\mathbf{A = B_1 \oplus E_1} \quad (33)$$

$$\mathbf{A = B_2 \oplus E_2} \quad (34)$$

$$\mathbf{A = B_3 \oplus E_3} \quad (35)$$

The correct data vector **A** given by (33), (34) and (35) should be same which indicates that the constraints (9), (10) and (11) are satisfied and the recovery of **A** is successful.

D. Algorithm to determine the correct A at the receiver

Since, we are using three consecutive erroneous vectors to determine the correct data vector **A**, we call our proposed method as FEC_3V Algorithm. The algorithm is implemented at the receiver.

Algorithm FEC_3V**.**

INPUTS : Three consecutive erroneous vectors $\mathbf{B_1}$, $\mathbf{B_2}$ and $\mathbf{B_3}$**.**

The size N of all vectors.

ASSUMPTION: Constraints (9), (10) and (11) should be Satisfied.

OUTPUT: Correct data vector **A** and Result *Success Flag.*

1. Receive $\mathbf{B_1}$. Check CRC.
   If no error, $\mathbf{A = B_1}$. Go to step 8.
   Else, store $\mathbf{B_1}$. Request source to retransmit **A**.
2. Receive $\mathbf{B_2}$. Check CRC.
   If no error, $\mathbf{A = B_2}$. Go to step 8.
   Else, store $\mathbf{B_2}$. Request source to retransmit **A**.
3. Receive $\mathbf{B_3}$. Check CRC.
   If no error, $\mathbf{A = B_3}$. Go to step 8.
   Else, store $\mathbf{B_3}$.
4. Calculate $\mathbf{F_{12}}$, $\mathbf{F_{13}}$ and $\mathbf{F_{23}}$ using (12), (13) and (14).
5. Calculate $\mathbf{E_1}$, $\mathbf{E_2}$ and $\mathbf{E_3}$ using (30), (31) and (32).
6. Calculate **A** using (33), (34) and (35).
7. Check that all the three vector values of **A** are equal.
   If no, set *Success Flag* **False.** Go to step 8.
   Else, set *Success Flag* **True.**
8. Exit.

E. Example

Example 1. The inputs $\mathbf{B_1}$, $\mathbf{B_2}$ and $\mathbf{B_3}$ are taken as,

$\mathbf{B_1}$ = [1 0 0 0 1 0 0 1 0 0]

$\mathbf{B_2}$ = [1 0 1 0 0 1 0 0 0 0]

$\mathbf{B_3}$ = [0 0 1 0 0 0 0 1 1 1]

Now, $\mathbf{F_{12}}$, $\mathbf{F_{13}}$ and $\mathbf{F_{23}}$ calculated using (12), (13) and (14) are found to be,

$\mathbf{F_{12}}$ = [0 0 1 0 1 1 0 1 0 0]

$\mathbf{F_{13}}$ = [1 0 1 0 1 0 0 0 1 1]

$\mathbf{F_{23}}$ = [1 0 0 0 0 1 0 1 1 1]

Then, $\mathbf{E_1}$, $\mathbf{E_2}$ and $\mathbf{E_3}$ are calculated using (30), (31) and (32) to get,

$\mathbf{E_1}$ = [0 0 1 0 1 0 0 0 0 0]

$\mathbf{E_2}$ = [0 0 0 0 0 1 0 1 0 0]

$\mathbf{E_3}$ = [1 0 0 0 0 0 0 0 1 1]

Then **A** is found using (33), (34) and (35).Vector **A** is found to be same in all the three cases as,

A = [1 0 1 0 0 0 0 1 0 0]

F. Discussion

In algorithm FEC_3V, the calculation of **A** does not depend on the values of m_1, m_2 and m_3 so long as constraints (9), (10) and (11) are satisfied. That means, the algorithm works correctly irrespective of the number of errors present in the received vectors $\mathbf{B_1}$, $\mathbf{B_2}$ and $\mathbf{B_3}$. However, the sum, $(m_1 + m_2 + m_3) \leq N$. Otherwise, errors among $\mathbf{B_1}$, $\mathbf{B_2}$ and $\mathbf{B_3}$ overlap each other and the constraints (9), (10), (11) do not hold true.

1) Probability of overlapping of errors: Consider the case of a single error in each of $\mathbf{B_1}$, $\mathbf{B_2}$ and $\mathbf{B_3}$. That is $m_1 = m_2 = m_3 = 1$. Then, probability of a single bit error in vector $\mathbf{E_1}$ at location u, for $1 \leq u \leq N$, represented by $Q(u, \mathbf{E_1})$ is,

$$Q(u, \mathbf{E_1}) = Np \tag{36}$$

where, p is the Bit Error Ratio (BER) and N is the total number of bits in $\mathbf{E_1}$. We have approximated (1−p) by 1. Similarly, $Q(v, \mathbf{E_2}) = Np$. Here, v is one of the locations out of N. But v and u need not be same. Therefore, the probability it occurs at the same location as in $\mathbf{E_1}$ is given by,

$$Q(v = u, \mathbf{E_2}) = Np/N = p \tag{37}$$

From (36) and (37), probability that the single error occurs at the same location (overlap) in $\mathbf{E_1}$ and $\mathbf{E_2}$ is given by,

$$Q(\text{overlap}) = Q(u, \mathbf{E_1})Q(v = u, \mathbf{E_2}) = (Np)p = Np^2 \tag{38}$$

Since, we have 3 pairs to be considered for overlap, the probability that an overlap occurs among the three pairs is given by,

$$P(\text{single error bit overlap in } \mathbf{E_1}, \mathbf{E_2} \text{ and } \mathbf{E_3}) = 3Np^2 \tag{39}$$

Therefore, probability of no overlap or success represented by

$$Q(\text{success}) = 1 - 3Np^2 \tag{40}$$

Since p is generally in the range 10^{-2} to 10^{-4} for ordinary noisy channels, the value $3Np^2$ is low. Q(success) is very high. Probabilities of overlap can be determined for different values of m_1, m_2 and m_3. Thus probability of no overlap is very high.

LINK QUALITY COMPARISON

The link quality performance of our proposed method is compared with that of a simple Stop and Wait ARQ (SW-ARQ) method. We use the Required Number of Packets (RNP) [6], [7], [8] metric to measure the link quality performance.

A. Required Number of Packets (RNP) Metric

In an ARQ system, RNP is defined [6] as the Average number of retransmissions of the same packet until the packet received is error free. For good performance, RNP should be as low as possible. Let PSP be the Packet Success Probability between the transmitter and the receiver. Then we use the geometric distribution [9] for the number of attempts required for the first success. Then RNP for the simple ARQ system is the mean of the number of additional attempts and it is given by,

$$\text{RNP}_{\text{ARQ}} = \frac{1}{\text{PSP}} \tag{41}$$

In terms of p, the PSP, which is the probability no error in all the N bits of the frame, is given by,

$$\text{PSP} = (1-p)^N \tag{42}$$

From (41) and (42),

$$\text{RNP}_{\text{ARQ}} = (1-p)^{-N} \tag{43}$$

Since p is small compared to 1, using binomial expansion, (43) is approximated as,

$$\text{RNP}_{\text{ARQ}} = 1 + p*N \tag{44}$$

In our proposed FEC_3V, the number of retransmissions is 2 when there is no overlap error. If there is an error additional retransmission takes place. Using, geometric distribution, the average number of additional retransmissions is,

$$\text{RNP}_{\text{FEC\_3V}} = \frac{1}{Q(\text{success})} \tag{45}$$

where, Q(success) is given by (40). From (40) and (45),

$$\text{RNP}_{\text{FEC\_3V}} = \frac{1}{(1 - 3Np^2)} \tag{46}$$

Using the binomial expansion, (46) yields $\text{RNP}_{\text{FEC\_3V}}$ approximately as,

$$\text{RNP}_{\text{FEC\_3V}} = 1 + 3Np^2 \tag{47}$$

Equation (48) holds good for single errors in $\mathbf{E_1}$, $\mathbf{E_2}$ and $\mathbf{E_3}$.

Since P*N in (44) and $3NP^2$ in (47) are very small compared to 1, to overcome the masking effect of 1, while comparing RNP's, we consider the values of the two terms exceeding 1 as,

$$\Delta\text{RNP}_{\text{ARQ}} = \text{RNP}_{\text{ARQ}} - 1 = p*N \tag{48}$$

$$\Delta\text{RNP}_{\text{FEC\_3V}} = \text{RNP}_{\text{FEC\_3V}} - 1 = 3Np^2 \tag{49}$$

Example 2. Here, we have taken N =200 and p is increased starting from 10^{-7} up to $10^{-3.5}$ in steps of $10^{0.5}$ Then, RNP_{ARQ} and $\text{RNP}_{\text{FEC\_3V}}$ are calculated. Since the calculated values vary over a wide range, the graph is plotted in the logarithmic scale as shown in Fig. 2.

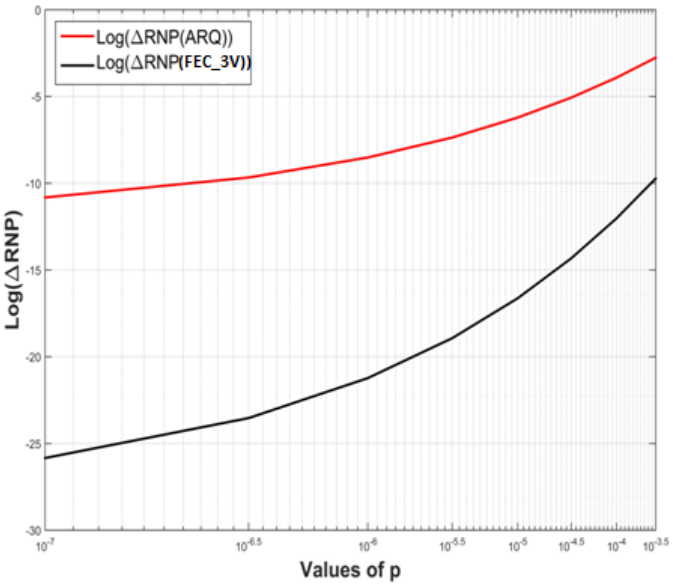

Figure 2. Variations of RNP's with p

B. Percentage reduction in RNP

The percentage reduction in RNP is calculated using the formula,

$$\%\text{Red} = \left(\frac{\text{RNP}_{\text{ARQ}} - \text{RNP}_{\text{FEC\_3V}}}{\text{RNP}_{\text{ARQ}}}\right) * 100 \qquad (50)$$

Example 3. Percentage reduction value, % Red is calculated for different values of p starting from p = 10^{-5}, incrementing in multiples of 2 as, 2*p, 4*p ,..,128*p and for N = 300, 400 and 500. The result is shown in Table 2. The corresponding values are shown in the bar graph of Fig. 3. From the result we can see that the percentage reduction increases as N increases.

Table 2: Variation of %Red with respect to p. Initial p =10^{-5}

Value of p	%Red		
	N = 300	N = 400	N = 500
p	0.2991	0.3984	0.4975
2*p	0.5964	0.7936	0.9900
4*p	1.1856	1.5747	1.9605
8*p	2.3432	3.1005	3.8452
16*p	4.5780	6.0142	7.4039
32*p	8.7507	11.3447	13.7799
64*p	16.0765	20.3743	24.1959
128*p	27.6391	33.8468	38.8745

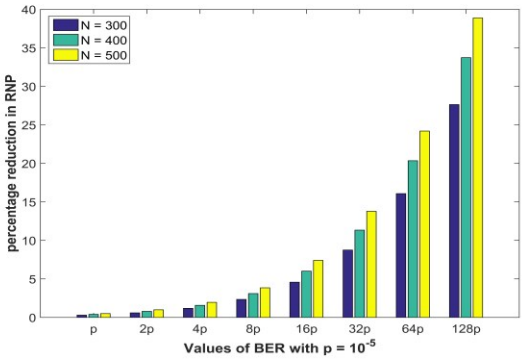

Figure 3: Variations of % reduction in RNP's with p

Thus it is found that our proposed method saves the number of retransmissions and hence reduces the energy consumption in transmitters. This in turn increases the battery life of sensor nodes.

Compared to other types of forward error correction codes like Hamming codes, Reed-Solomon, Turbo, LDPC's and convolution codes [10], our method has less computational complexity.

CONCLUSIONS

A new method FEC_3V is described for multiple forward error correction in the link layer. FEC_3V is a new ARQ flow control mechanism. When the channel BER is high, with only two retransmissions, FEC_3V can correct a reasonably large number of bit errors per frame, the method uses XOR, simple arithmetic operations, and CRC checking. The method is highly effective when the size of the data per frame is relatively high.

REFERENCES

[1] Shu Lin, Daniel J. Costello. "*Error Control Coding: Fundamentals and Applications"*. Englewood cliffs, NJ: Prentice-Hall, 1982.

[2] Jung-Fu (Thomas) Cheng, "Coding Performance of Hybrid ARQ Schemes," IEEE transactions on communications, vol. 54, no. 6, pp. 1017-1029, June 2006.

[3] C. Lott, "Hybrid ARQ: Theory, State of the Art and Future irections," IEEE Information Theory Workshop on Information Theory for Wireless Networks, July 2007.

[4] D. Chase, "A combined coding and modulation approach for communications over dispersive channels," *IEEE Trans. Commun.*, vol. COM-21, no. 3, pp. 159–174, Mar. 1973

[5] D. Chase, "Code combining - A maximum-likelihood decoding approach for combining an arbitrary number

of noisy packets," IEEE Trans. on Comm., vol. 33, No. 5, pp. 385-393, May 1985.

[6] A. Cerpa, J. L. Wong, M Potkonjak and D. Estrin. "Temporal Properties of Low Power Wireless Links: Modeling and Implications on Multi-Hop Routing," In MobiHoc'05, Urbana-Champaign, IL, May 2005.

[7] D. Lal, A. Manjeshwar, F. Herrmann, E. Uysal-Biyikoglu, A.Keshavarzian. "Measurement and Characterization of Link Quality Metrics in Energy Constrained Wireless Sensor Networks," In *GLOBECOM'03*, pp. 446-452. San Francisco, 2003.

[8] Tao Liu, Ankur Kamthe, Lun Jiang and Alberto Cerpa, "Performance Evaluation of Link Quality Estimation Metrics for Static Multihop Wireless Sensor Networks," Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference. pp. 1-9, Rome, June 2009.

[9] Geometric distribution - Wikipedia, the free encyclopediaen.wikipedia.org/wiki/Geometric_distribution

[10] Mohammad Rakibul Islam, "Error Correction Codes in Wireless Sensor Network: An Energy Aware approach," International Journal of Computer and Information Engineering, pp. 59-64, 2010.