ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 6, August 2012

INTERLEAVED FORWARD ERROR CORRECTING (FEC) CODES BASED ON BINAR SHUFFLE ALGORITHM (B.S.A)

Phani deep challa, P.Aaravind, Vemuri Seshagiri Rao

***Abstract*— Forward Error Correcting (FEC) is one of the technique is used for controlling errors without sending any retransmission to sender. Example of forward error correcting codes are hamming, lower-density parity-check (LDPC), tail-biting convolutional, turbo codes. Many wireless systems such as worldwide interoperability for microwave access (WiMAX), long term evolution (LTE) and enhanced data rates for the GSM evolution (EDGE) use forward correcting codes to sending messages. The main aim of this project is send the message from source to destination correctly (without losing any packets) and without sending any retransmission to sender. Here Binar Shuffle Algorithm is used to shuffle data in the interleaving process then burst errors are convert into random errors. The decoder detect and correct the errors very easy by using Belief Propagation (BP) Algorithm. Than performance also very high compared to with-out using interleaving Algorithms.**

Index Terms—Forward Error Correctig Codes, Binar shuffle Algorithm, Interleaving process, Belief Propagation (B.P) Algorithm, Convolutional codes.

Manuscript received July, 2012

Phani Deep Challa, *M.Tech CSE 2'nd Year Student, (Computer Sciencce and Engineering),J.N.T.U Kakinada/ Gayatri Vidya Parishad Engineering College, Vijayawada, India,+918500263931.*

P.Aaravind, *Associate Professor, Computer Sciencce and Engineering, J.N.T.U Kakinada/ Gayatri Vidya Parishad Engineering College, Visakhapatnam, India, +918912739507.*

Vemuri Seshagiri Rao, *Professor & HOD, Computer Sciencce and Engineering, J.N.T.U Kakinada/ Gayatri Vidya Parishad Engineering College, isakhapatnam, India,+918885043413.*

I.NTRODUCTION

Interleaving process is very important in the digital communication. Due to the interleaving process if any packets are loosed than regain the loosed-packets are very easy on decoding side. Our research system detect and correct more errors compare to previous system (i.e. without using interleaving process). Our proposed system uses the *Binar Shuffle Algorithm* for interleaving process than all burst errors convert into random errors.

All Rights Reserved © 2012 IJARCET

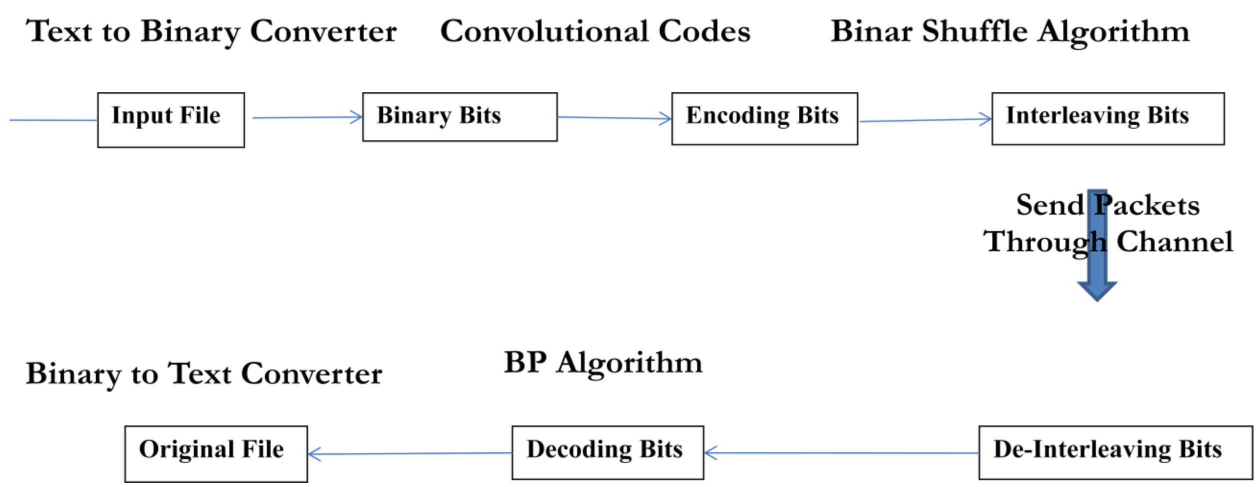


Figure 1: Overall System Architecture

Figure 1 shows the Overall System Architecture in wireless system. Text to Binary Converter is used for to covert text file into binary file. The encoding channel is used for adding redundant bits to original bits by using convolutional codes. Then *Binar Shuffle Algorithm* is used for shuffling bits in the interleaving process and send the packets through the channel. Above process is done at the sender side.

After the sender side all remaining process is done at the receiver side. De-interleaving process should be done after receiving bits. If any packets are loosed, our system should Detect and Correct errors very easy. For Detect and Correct the errors proposed system uses the Belief Propagation Algorithm on the decoding channel.

The packet transport service provided by representative packet-switched networks, including IP networks, is not reliable and the quality-of-service (QoS) cannot be guaranteed. Packets may be lost due to buffer overflow in switching nodes, be discarded due to excessive bit errors and failure to pass the cyclic redundancy check (CRC) at the link layer, or be discarded by network control mechanisms as a response to congestion somewhere in the network. Forward error correction (FEC) coding has often been proposed for end-to-end recovery from such packet losses.

The rest of this paper organized as follows. In Section II, forward Error Correcting codes. In Section III, describe about convolutional codes with necessary notations and definitions. In Section IV and V, describe the interleaving process to shuffle the data and decoding channel by BP Algorithm. In Section VI, shows conclusion of the system.

All Rights Reserved © 2012 IJARCET

II. FORWARD ERROR CORRECTING CODES

forward error correction (FEC) or channel coding[1] is a technique used for controlling errors in data transmission over unreliable or noisy communication channels. The central idea is the sender encodes their message in a redundant way by using an error-correcting code (ECC). FEC is accomplished by adding redundancy to the transmitted information using a predetermined algorithm. Each redundant bit is invariably a complex function of many original information bits. The original information may or may not appear in the encoded output; codes that include the unmodified input in the output are systematic, while those that do not are nonsystematic. An extremely simple example would be an analog to digital converter that samples three bits of signal strength data for every bit of transmitted data. If the three samples are mostly all zero, the transmitted bit was probably a zero, and if three samples are all one, the transmitted bit was probably a one. FEC codes or channel codes have become an inevitable in wireless based digital communication systems. That is, by allowing a system to operate at a lower signal to noise ratio than would otherwise be the case, a desired quality of service over a link can be achieved within a transmit power or antenna gain constraints of the system. This property of error correcting codes is often referred to as 'Power Efficiency'[2].

Forward Error Correction (FEC) is a type of error correction which improves on simple error detection schemes by enabling the receiver to correct errors once they are detected. This reduces the need for retransmissions.

FEC works by adding check bits to the outgoing data stream. Adding more check bits reduces the amount of available bandwidth, but also enables the receiver to correct for more errors.Forward Error Correction is particularly well suited for satellite transmissions, where bandwidth is reasonable but latency is significant.

III. CONVOLUTIONAL CODES

First introduced by Elias in 1955 [3], binary convolutional codes are one of the most popular forms of binary error correcting codes that have found numerous applications [4]. A convolutional code is called tail-biting when its codewords are those code sequences associated with paths in the trellis that start from a state equal to the last m bits of an information vector of k data bits. Many efficient algorithms have been proposed for decoding tail-biting convolutional codes such as the Viterbi and MAP algorithms. As shown below, we represent the tail-biting convolutional code by its generator and parity-check matrices in order to apply the BP algorithm directly. Convolution codes are a popular class of coders with memory, i.e., the coding of an information block is a function of the previous blocks [5]. A Convolutional code is a type of error-correcting code in which **(a)** Each m-bit information symbol (each m-bit

All Rights Reserved © 2012 IJARCET

string) to be encoded is transformed into an n-bit symbol, where m/n is the code rate (n ≥ m) and **(b)** The transformation is a function of the last k information symbols, where k is the constraint length of the code. Convolutional codes are often used to improve the performance of digital radio, mobile phones, satellite links, and Bluetooth implementation [6]. Convolution codes involve simple arithmetic operations and therefore they are easily implemented. If a block code is used for error detection, only simple integer division is needed; however, decoding block codes or convolution codes for error correction is much more tedious [7]. For block codes, an iterative algorithm is often used to correct the errors. Error correction algorithms become quite complex for long codes with large error correction capability; especially, for non-binary codes.

IV. INTERLEAVING PROCESS

Interleaving process is one of the technique to handle more bit rate errors by using shuffle algorithms. In this system, shuffle the data by using binar shuffle algorithm than burst errors are convert into random errors. order to increase performance. It is used in Time-division multiplexing (TDM) in telecommunications, Computer memory and disk storage [8].

Example Usage:

Error-free code words: aaaabbbbccccddddeeeeffffgggg

Interleaved:
abcdefgabcdefgabcdefgabcdefg

Transmission with a burst error: abcdefgabcd____bcdefgabcdefg

Received code words after deinterleaving: aa_abbbbccccdddde_eef_ffg_gg

In each of the codewords aaaa, eeee, ffff, gggg, only one bit is altered, so our one-bit-error-correcting-code will decode everything correctly. Of course, latency is increased by interleaving because we cannot send the second bit of codeword aaaa before awaiting the first bit of codeword gggg.

Burst errors: A burst error or error burst is a contiguous sequence of symbols, received over a data transmission channel, such that the first and last symbols are in error and there exists no contiguous subsequence of m correctly received symbols within the error.

Random errors: Random errors are errors in measurement that lead to measurable values being inconsistent when repeated measures of a constant attribute or quantity are taken. The word random indicates that they are inherently unpredictable, and have null expected value, namely, they are scattered about the true value, and tend to have null arithmetic mean when a measurement is repeated several times with the same instrument.

All Rights Reserved © 2012 IJARCET

Binar Shuffle Algorithm

Shuffling is a process of re-ordering data elements of a sequence from an initial permutation into
a random arrangement of an arbitrary permutation. A shuffle algorithm scrambles data elements
into a random placement without any apparent organizing key evident. Shuffling is a process of re-ordering data elements of a sequence from an initial permutation into a random arrangement of an arbitrary permutation. A shuffle algorithm [9] scrambles data elements into a random placement without any apparent organizing key evident.

1. 1.The probabilistic definition of shuffling is:
   Given a sequence *S* of *N* records R0, R1, ... , Rn-2, Rn-1 that are arranged in a permutation: p(0)p(1) ... p(n-2)p(n-1).

2. The sequence S is considered shuffled if for the k possible selection of any two records Ri and Rj.
   where i ≠ j, that the probability of Ri ≤ Rj is equal to Ri > Rj for 0 < k ≤ N!.

3. For the kth possible selection, the probability Pk is:
   Pk(Ri ≤ Rj) = Pk(Ri > Rj) where i ≠ j and 0 < k ≤ N!.

4. Thus after one or many selections, that any two unique records that the probability is equally likely for lesser or greater relation. No matter how many times two distinct records are selected, the overall probability of lesser or greater remains equal—there is no bias.

5. For the kth possible selection, the probability Pk is:
   Pk(Ri ≤ Rj) = 1.0 where i < j and 0 < k ≤ N!

For any two different records in a sorted sequence where the records are in increasing positions it
is always true that the records maintain the lesser than relation.
It is equally likely for lesser or greater for any kth number of selections from the sequence.
Shuffling is a random permutation of the records in a sequence, thus for the selection of records the two relations are equally likely for one, two, three, or many selection

All Rights Reserved © 2012 IJARCET

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 6, August 2012

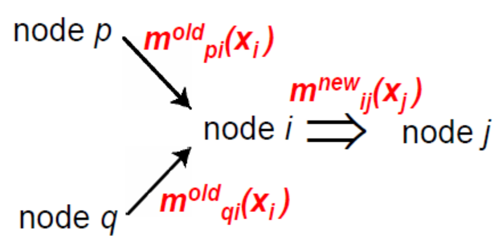
V. DECODING PROCESS

Belief Propagation Algorithm is used for Decode data. Until recently, most known decoding algorithms for convolutional codes were based on either algebraically calculating the error pattern or on trellis graphical representations such as in the MAP and Viterbi algorithms. With the advent of turbo coding [10][14], a third decoding principle has appeared: iterative decoding. Iterative decoding was also introduced in Tanner's pioneering work [11], which is a general framework based on bipartite graphs for the description of LDPC codes and their decoding via the belief propagation (BP) algorithm. Here all burst errors are converted into random errors. So that decoding process is very easy by using the belief propagation Algorithm.

Technique invented in 1982[Pearl] to calculate marginals in Bayes nets. Also works with MRFs, graphical models, factor graphs. Exact in some cases, but approximate for most problems. Can be used to estimate marginals, or to estimate most likely states (e.g. MAP).

Belief propagation, also known as Sum-product message passing is a message passing algorithm for performing inference on graphical models, such as Bayesian networks and Markov random fields. It calculates the marginal distribution for each unobserved node, conditional on any observed nodes. Belief propagation[12][13] is commonly used in artificial intelligence and information theory and has demonstrated empirical success in numerous applications including low-density parity-check codes, turbo codes, Convolutional codes, free energy approximation, and satisfiability.

1. To update message from i to j, consider all messages flowing into i (except formessage from j):



2. The messiest equation in this tutorial:

$$m_{ij}^{new}(x_j) = \sum_{x_i} f_{ij}(x_i, x_j) \underbrace{g_i(x_i) \prod_{k\in Nbd(i)\backslash j} m_{ki}^{old}(x_i)}_{h(x_i)}$$

3. Messages (and unary) factor on RHS multiply like independent likelihoods update equation has this form:

$$m_{ij}^{new}(x_j) = \sum_{x_i} f_{ij}(x_i, x_j) h(x_i)$$

Note: given a pair of neighboring nodes, there is only one pairwise interaction but messages
flow in *both* directions.

4. Define pairwise potential so that we can use the message update equation in both directions (from *i* to *j* and from *j* to *i*) without problems:

$$f_{ij}(x_i, x_j) = f_{ji}(x_j, x_i)$$

5. By the way, this isn't the same as assuming a symmetric potential, i.e.

All Rights Reserved © 2012 IJARCET

$$f_{ij}(x_i, x_j) = f_{ij}(x_j, x_i)$$

6. In practice one usually normalizes the messages to sum to 1, so that

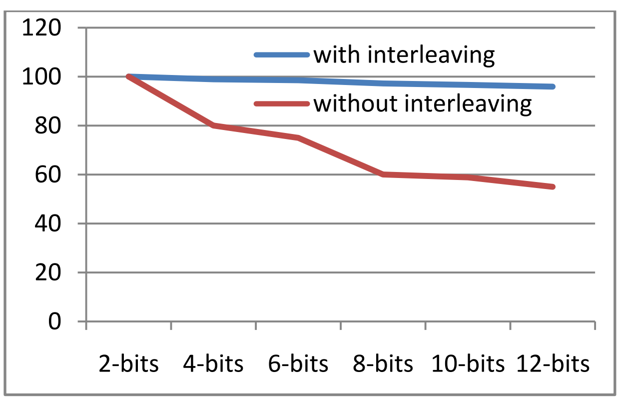
$$\sum_{x_j} m_{ij}(x_j) = 1$$

Useful for numerical stability (otherwise overflow/ underflow likely to occur after enough message updates).

VI. SIMULATION RESULTS

The simulation results between interleaving process and without interleaving process. In the graph packet loss in bits indicated on X-axis and performance in percentages (%) on Y-axis. Blue line represents the with interleaving process. Red line represents the without interleaving process. All the above process is shown in below Table1.

Table1: Performance of with interleaving and without interleaving.



Both performance almost equal at 2-bit errors. While number of loosing bits increasing, performance of without process gradually decreasing compared to the interleaving process.

VII. CONCLUSION

In this paper mainly concentrate on interleaving and decoding process. The interleaving process is done by binary shuffle Algorithm. So that all burst errors are converted into random errors. Due to this process, decoding process became very easy by using belief propagation Algorithm. Due to the interleaving process in Foreword Error Correcting codes, transmit the messages more efficiently in wireless network.

REFERENCES

[1] X. Yu, J. Modestino, and I. Bajic, "Performance analysis of the efficacy of packet-level FEC in improving video transport over networks," in Proc. IEEE Int. Conf. Image Process. (ICIP '05), pp. II-177–II-180.

[2] Dan Jurca et al, "Forward Error Correction for Multipath Media Streaming" IEEE Transactions on circuits and systems for video technology, vol. 19, no. 9, pp 1315-1326, September 2009.

[3] P. Elias, "Coding for Noisy Channels," IRE Conv. Rec., vol. 3, pt. 4, pp. 37–46, 1955.

[4] J. Viterbi, "Convolutional Codes and their Performance in Communication Systems," IEEE Trans. On Commun., vol. 19, no. 15, pp. 751-772, 1971.

[5] 3GPP TSG-SA4#31 Tdoc S4-040348 May 17-21, 2004, Montreal, Canada, Simulation Guidelines for the Evaluation of FEC Methods for MBMS Download and Streaming Services.

All Rights Reserved © 2012 IJARCET

[6] Frenger, P., P. Orten, and T. Ottosson, "Convolution Codes with Optimum Distance Spectrum," IEEE Communications Letters, vol. 3,pp. 317-319, November 1999.

[7] Ufuk DEMIR and Ozlem AKTA$ "Raptor versus Reed Solomon Forward Error Correction Codes" Proceedings of the Seventh IEEE International Symposium on Computer Networks (ISCN'06) pp 264-269, 2006

[8] L. Yin, J. Lu, K. Ben Letaief and Y. Wu, "Burst-error correcting algorithm for Reed- Solomon codes", ElectronicsLetters, Vol. 37 No. 11, May 25, 2001, pp. 695- 697.

[9] Binar Shuffle Algorithm: Shuffling Bit by Bit by William F. Gilreath August 2008.

[10]C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," IEEE Intl. Conf. on Commun., vol. 2, Geneva, Switzerland, pp. 1064-1070, 1993.

[11]R. M. Tanner, "A recursive approach to low complexity codes," IEEE Trans. Inform. Theory, vol. 27, no. 5, pp. 533-547, 1981.

[12]"A Tutorial Introduction to Belief Propagation" by James Coughlan .

[13]A. Refaey, S. Roy, and P. Fortier, "On the Application of BP Decoding to Convolutiona and Turbo Codes," Asilomar Conference on Signals, Systems, and Computers, Nov. 2009.

[14]C. Poulliat, D. Declercq, and T.Lestable, "Efficient Decoding of Turbo Codes with Nonbinary Belief Propagation," EURASIP Journal on Wireless Communications and Networking, vol. 2008, no. 473613, 2008.

Phani Deep Challa received the B.Tech Degree in CSE From the Gudlavalleru College Of Engineering Affiliated to Jawaharlal Nehru Technical University, Kakinada in 2010. Currently Studying M.Tech in C.S.E From the Gayatri Vidya Parishad College of Engineering(Autonomous), Affiliated to Jawaharlal Nehru Technical University, Kakinada in 2010-12. Fields of interest is Networking, Programming logics and Web-Designing.

P.Aravind received the B.Tech and M.Tech Degree in CSE From the Gayatri Vidya Parishad College of Engineering(Autonomous), Affiliated to Jawaharlal Nehru Technical University, Kakinada. Currently working as Associate Professor in Gayatri Vidya Parishad Engineering College. He has the 5.5years Teaching Experience in Gayatri Vidya Parishad College of Engineering(Autonomous). Fields of interest is Networking, Dataminig, Computer Graphics.

Vemuri Seshagiri Rao received the B.E Degree From the Andhra University, Visakhapatnam. He did M.S in IE, Purdue University, West Lafayette, Indiana, USA and received Ph.D Degree in System Engineering From University of Virginia, Charlottesville, Virginia, USA . He is the member of Institute of Electrical and Electronics Engineers (IEEE). He received the Team of Distinction Award at BAE Systems in 2005. He research has published in leading 9 International Journals and in Conferences. He received JTRS "Cluster One" Proposal Aard and Chairman's Bronze Award for highest customer satisfaction in the development of Interface Control Documents in BAE Systems.

All Rights Reserved © 2012 IJARCET

All Rights Reserved © 2012 IJARCET