International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 5, May 2014

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

Error Detection and Correction for ATM Technology

Noor Bahaaldin[1], **Ergun Erçelebi**[2]

Technology University, Iraq[1]

Electric- Electronics Department, Gaziantep University, Turkey[2]

Abstract: In this paper, we develop Reed-Solomon (RS) detection and correction implementation to work in an ATM network. The investigating tool selected is MATLAB, a new method that utilizes Berlekamp Massey Algorithm (BMA) in Reed Solomon (RS) decoder to reduce complexity is proposed for error detection and correction. BMA works with the individual coefficient of the polynomial instead of the entire polynomial as a whole, to correct efficiently both errors and erasures. We have obtained an optimization in ATM network with Forward Error Correction (FEC) method. FEC corrects the corrupted data rather than retransmitting it again. We have evaluated the performance of the proposed method for the random integer, information, and speech signal. The results show that the errors have been detected and corrected successfully. To obtain best bit error rate, the proposed method has been evaluated for different code rates. To assess the performance of proposed method we have compared it with other well-known code techniques. The comparison results are given in terms of BER versus SNR (Signal to Noise Ratio). The comparison results show that the proposed method is superior to others.

Index terms: Reed-Solomon (RS), Forward Error Correction (FEC), Berlekamp-Massy algorithm (BMA), Chien Search

I. INTRODUCTION

ATM (Asynchronous Transfer mode) is a superb-speed communication technique reached nearly a cosmopolitan admittance in the latter ages. ATM gives a superb-bandwidth, low-tardiness multiplexing/ switching (MUX/SW). A cell is ATM (MUX/SW)'s basic-unit. Its length=53 bytes: 48 data-bytes, 5 control-bytes such as a virtual-path-identifier (VIP), virtual-channel-identifier (VCI) and a cyclic-redundancy-checksum (CRC) for header error control. End-systems reconcile the difference if given-probability << requested-probability.

ARQ (Automatic Repeat request) and FEC (Forward Error Correction) are substantial mechanisms for Reliability's Amelioration. The data retransmission/closed-loop mechanism is the ARQ-precept in case of reception's errors. The state-information's exchange between dispatcher and remit-tee is what ARQ needs. A delay of 1 round-trip-time is added upon each retransmission. So, for low-latency services, ARQ may not be workable in such a field. Therefore, FEC is the ersatz solution. It averts the ARQ's blemishes and convenient for high-bandwidth-delay services. By Redundant-Information transmission with the Original Ones, FEC can rebuild the data again in case of data-loss. FEC keeps sufficient; because of redundant-information amount's smallness.

Larry S. Reed & Solomon had contrived the R-S codes in 1960 [1]; which are efficacious in deep-fades channel and considered as a structured sequence to be useful immensely in FEC for assortment communications to their error-correction-strength in burst errors. They also wrote, "Polynomial codes over certain fields" [2].

FEC is exemplary RS-application code, shown in Fig. 1. The original-data are transmitted to the remitter with the added redundant bits (parity-symbols) by FEC. Errors-along RS-decoding' mechanism can be altered to correct errors and erasures [1] [3] (when a corrupted-symbols' position is known, an erasure occurs). This modality of error detection/correction is utilized excessively in data communication services (e.g. Digital Video Broadcast (DVB), optical carriers like (OC -129). RS-characteristics control the corrections of error's type and numbers. The experimental system model provides error detection and correction using FEC. The message is encoded and decoded using error- correcting Reed-Solomon codes.

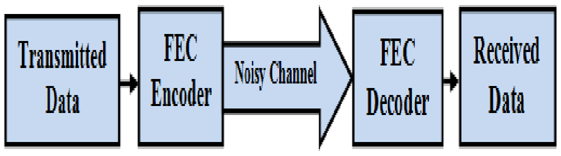


Fig. 1. Forward error correction concept

II. REED SOLOMON THEORY

RS-FEC became solemn in telecommunications. Most Reed-Solomon schemas are methodical. This intents that the segment of code word output includes a real form with input data. The Reed-Solomon code is an apparent code, so the decoders continue to operate if the symbols of the track have been inverted thereabouts through the line. The result will be original-data's complement. However, if RS is used to fill virtual zero, then it will lose the transparency. Therefore, before RS decoding, the sense of the data must be resolved [2].

RS is a block-based code that presented as RS (n, k) shown in Fig. 2. Over a Galois field GF (2^m). The size of the code word (the block length) with the unit of symbols are represented by the variable 'n', 'k' represent the number of information symbols, '2t' represent the number of parity symbols, and each symbol contains m-bits (symbol size).

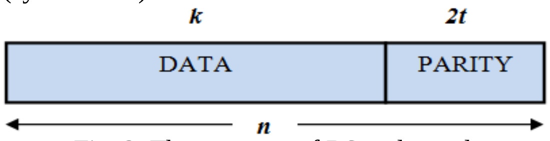


Fig. 2. The structure of RS codeword

The symbol-size (m) and code word-size (n) have a correlation like $n=2^m-1$; that is, in one code word which includes the one with all zeroes there exist one symbol 2^m-1 for m-bits. The maximum number of correctable random errors or the error-correction capability is given by this relation t = (n-k) /2. The code minimum distance can be calculated by relation $d_{min}=n-k+1$. For example, the size of the RS (255,239), 255 is the symbols and 239 is data symbols' numbers k. So depending on these values the decoder can correct the symbol errors up to t = (255-239) / 2 = 8 symbols.

RS-encoder adds extra "redundant" bits/ digital-data block. Errors take place throughout transmission or storage for a number of reasons (for example noise or interference, scratches on a CD, etc.). The RS decoder processes each block and tries to correct errors and retrieve the original information [4] [3].

III. GALOIS FIELD

RS-code is defined over GF [3]. It has a limited-group of numbers; thus an element's arithmetic operation/group gives an element within that group; the way of newfangled-mathematics is handling the retrieved-numbers from several given initial-series by operators like (+, -, x, /) where a field is called the resulting-series.

The unlimited-fields create an obstacle in the computer's representation of fixed-word's length (e.g. int-value). The valuable characteristic of the Galois Fields is that the operation of the element's field will outcome in another one.

Since it is bounded as well, it's symbolized as a constant long bilateral-word. For any p (prime number) there's a GP (P). GF (p^m) an extension field of GF (p) (m = +integer), which are used in construction of RS codes. Extension-field elements symbolized as α. All non-zero 'element in GF (2^m) denoted by a power of α, the elements of the bounded field GF (2^m) are as follows [2]:

$$GF\ (2^m)=\{\ 0,\alpha^0, \alpha^1, \alpha^2, \ldots,\ \alpha^{2m-2}\} \qquad (1)$$

Each 2^m elements of GF (2^m) can be symbolized as a diverse-polynomial of degree m-1 or less. The value of maximum-order exponent represents the degree of a polynomial. All nonzero element of GF (2^m) symbolized as a Polynomial $\alpha^i(X)$, where at least 1 per *m* coefficients of $\alpha^i(X)$ nonzero. Fora $i=0,\ 1,\ 2\ldots 2^m\text{-}2$.

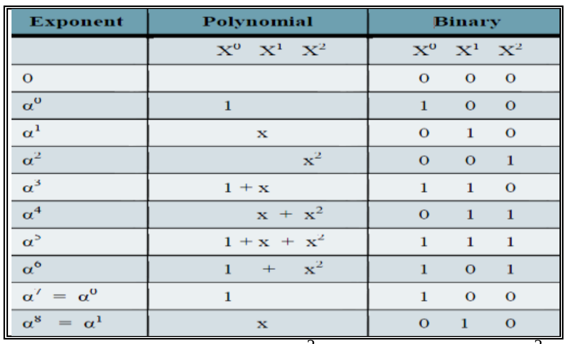
$$\alpha^i = \alpha^i(X) = \alpha^i,\ 0 + \alpha^i, 1X + \alpha^i,\ 2X^2 + \ldots + \alpha^i,\ m\text{-}1\ X^{m-1} \qquad (2)$$

If we considered m=3 where the finite field represented as GF (2^3) then RS-basis code will be RS (7, 3). Because each token has $\log_2$ (8) = 3 bits, the variables in RS code . $n=2^m-1=7$, K=3 (chosen to balance information l Numbers and parity tokens/ code word). Figure 3 shows the mapping of the zero elements with the seven elements $\{\alpha^{\ i}\}$ in the expression of the basic elements { X^1, X^1, X2} which explained by (1).

A. Operation in Galois Field

The Exclusive-OR (XOR) process using Galois Field (GF) represents the addition of two elements [5]. A (GF) multiplication is more risqué than any typical- arithmetic. Where the primeval –polynomial' module used to define (GF).

$$F(X)=1+x+x^3 \qquad (3)$$



Exponent	Polynomial	Binary		
	X^0 X^1 X^2	X^0	X^1	X^2
0		0	0	0
α^0	1	1	0	0
α^1	x	0	1	0
α^2	x^2	0	0	1
α^3	1 + x	1	1	0
α^4	x + x^2	0	1	1
α^5	1 + x + x^2	1	1	1
α^6	1 + x^2	1	0	1
$\alpha^7 = \alpha^0$	1	1	0	0
$\alpha^8 = \alpha^1$	x	0	1	0

Fig. 3. Elements for GF(2^3) with $F(X)=1+x+x^3$

IV. ENCODER

Because of the Encoder-methodology, the whole-block is readable to produce an output to latter-side with no-amendment. The parity token computation is completed, also can be given the full n symbols when the data symbols Kth has been read in. The aim of using the parity-tokens is to make a protracted-polynomial (n-grade-long– m(x) +p(x)) be partitioned using RS-G. P (Generator-Polynomial). By RS-G.P, the received information can be separated at the decoder And if the division's modulus=0 this yields 0-errors, otherwise there're errors. The procedure of the encoder which depends only on the current inputs is to divide message-sequence to 2-blocks: info.' block and redundancy/block. RS-symbols are GF's elements. Encoding is applied by annexing GF Polynomial-division's module to the information. The division achieved using LFSR (Linear Feedback Shift Register). LFSR is the primary computational RS-encoder's element. The calculation of the RS encoding is depending on the Finite Field operations. To define the Galois field, primitive polynomial used is

$$G(X)=1+X+X^3 \qquad (4)$$

We implemented a [7] [3] system, where (n,k) represents an out code word's length and an input word's length (k). It has a symbol's size (m) = three (n-k=2t), Where t is error correcting capability [4]. Therefore, the encoder has doubled error correcting capability. (7-3 = 2t) Therefore t = 2. The encoder forms a code word X^{n-k} m(X) + p(X), by means of the following equations

$$P(X)=X^n\text{-}k\ \ m(X)\ \ mod\ \ g(X) \qquad (5)$$

$$U(X)=p(X)+X^n\text{-}k\ \ m(X) \qquad (6)$$

Where *g(X)* is the generator polynomial, *m(X)* is the message, *P(X)* is parity, and *U(X)* is coded message polynomial. The RS generating polynomial's code takes this form:

$g(X) = g_0 + g_1 X + g_2 X^2 + \ldots + g_2 t\text{-}1X^{2t-1} + X^{2t}$ (7)

RS-G.P, Degree= parity symbols' Numbers. Generator polynomial has 2t = n - k = 4 roots, they are

$g(X) = (X-\alpha)(X-\alpha^2)(X-\alpha^3)(X-\alpha^4)$
$= X^4 - \alpha^3 X^3 - \alpha^0 X^2 - \alpha^1 X + \alpha^3$ (8)

In binary field+1= -1. In this state, the generator polynomial is given by

$g(X) = \alpha^3 + \alpha^1 X + \alpha^0 X^2 + \alpha^3 X^3 + X^4.$ (9)

In digital-HW, an encoder is represented by LFSR aalong with the interior-feedback-links which correspond to g(X). The operations involved are GF(+) and (×). To encode a 3-symbol-series in a typical-form with (7, 3), RS code described by g (X) is implemented using LFSR diagram shown in Fig. 4.

The encoder is a 2t tap shift register and has m-bits wide/register. The coefficients of RS generator polynomial are the steady-multiplier coefficient from α^3 to $\alpha^{(2t-1)}$ used-for multiplier's simplification, if required [7].

Initially, all-registers =0. An Addition occurs / clock-cycle between the symbol inside register with (feedback-symbol × steady-tap coefficient), then carry on to the following-register.

Feedback' value is final register's symbol on the next clock cycle. Finally, the values of the parity symbol are sitting at the register after reading all code word symbols and remains to shift them out one after another.

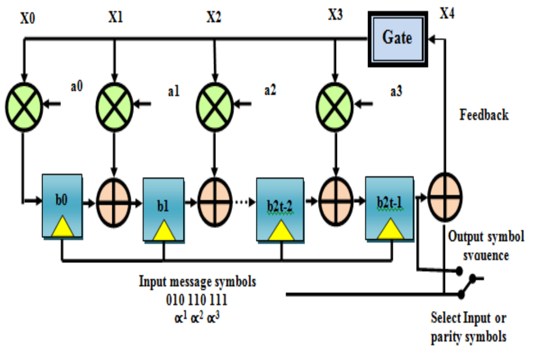


Fig. 4. Reed-Solomon encoder block diagram (LFSR)

V. DECODER

The decode operation takes several steps [2]; thus it's laborious. Throughout a noisy channel, the encoded data r (x) will become c (x) plus e (x) (error polynomial). It will be possible for the decoder to recover the corrupted message after computing e (x) and added it to the received message r (x), as given below:

$C(x) = r(x) + e(x) = c(x) + e(x) + e(x) = c(x).$ (10)

Where, the addition of e(x) to e(x) is equal to zero in Galois Field. A typical decoder follows the following stages:

1. Syndrome Calculation
2. Determine the error-location polynomial
3. Solving the error locator polynomial
4. Calculating the error Magnitude
5. Error Correction.

A. Syndrome Calculation

The determination of the data syndrome is the primary step for decoding the r (x); which then it will be divided using the generator polynomial to have a null value; if r (x) is a valid code word, and in order to assure that the code is precisely divisible by the generator, the parity should have to be added within the code word. The remainder represents the syndrome; which is calculated according to the 2t roots of the g(x) into r(x). The syndrome polynomial is usually represented as,

$$S_i = r(X)|_{X=\alpha i} = r(\alpha i) \quad i=1,\ldots,n\text{-}k. \tag{11}$$

The circuit below in the Figure (5) produces the S_i sequences (ith syndrome). The syndromes rely only on the errors.

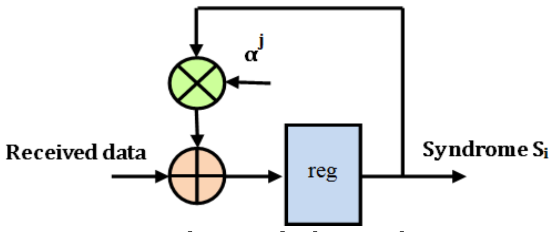


Fig. 5. Syndrome calculator architecture

B. Berlekamp-Massy Algorithm

In RS codes, the BMA is the most efficient algorithm in the decoding operation. It is impossible to solve the 2t simultaneous equations by the usual methods; since they are non-linear equations. Therefore, RS decoding algorithm is the distinct one in solving such equations. In addition, we have 2t unknown (the first t which is the error values and the second t which is the locations). According to the limitations of the study, we tend to consider only two commonly used RS decoding algorithms:

1) Berlekamp-Massey algorithm: it is considered an efficient computational method according to the number of operations in GF (2^m).
2) Euclidean algorithm: This one is less efficient, but is widely used in practice for its simplicity. The part of BMA is to compute the error-location polynomial's coefficients σ (X). After that, their roots are calculated by Chien search algorithm. If the σ (X) degree calculates by BMA larger than the value of it, then this indicates that it exceeded the maximum value of the correctable error and the code word is incorrect. Also, this RS decoding procedure is done when an error exists.

The Berlekamp algorithm is now qualified as:
1) The variables are initialized
2) K=0, L=0, $\wedge^{(0)}(X)=1$, T(X)=X, where $\wedge^{(0)}(X)$ is a BMA initial coefficient.
3) KK=KK+1. Compute discrepancy d(k)
4) $d(k) = S_k - \sum_{i=1}^{l} \wedge^{k-1} S_{k-i}$

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 5, May 2014

5) If d(k)=0
6) $\sigma^{i+1}(x) = \sigma(x)$, $l_{i+1} = l_i$
7) $\wedge^{(k)} = \wedge^{(k-1)}(x) - d^{(k)}T(x)$
8) If 2L = k then go to step 7
9) If L = k-L then keep continuing
10) $T(x) = \Lambda^{(k-1)}(x)/d^{(k)}$
11) T(x) = xT(x)
12) If k<2t go to step 2, otherwise the error's locator polynomial = $\wedge(x)$.

BMA role is to investigate a linkage-polynomial σ(i+1)(x) of minimal degree. Iteratively, it attempts to solve for the error locator polynomial.

C. Chien-Search

The zero's calculation of the error-finder-poly is applied in Chien-Search, which is the most efficient algorithm that is immediately searching all the roots of the calculated coefficients using the BMA. By computing, the error positions are the inverse of these roots [5].

$$\wedge(x) = 1+\wedge_1 x+\ldots..+\wedge_t x^t \qquad (12)$$

The Chien search consequently computes Λ(x); in order to evaluate the polynomial's roots in all the elements of the field α^0 α^1 α^2 α^3 … α^N. The roots, thus obtained will now point to the error locations in the received message. RS decoding generally employ the Chien search scheme to implement the same, if the result of substitution of its value in the polynomial evaluates to zero. Chien search is a brute force approach for guessing the roots and adopts direct substitution of elements in the Galois field, until a particular i from i=0, 1… (n-1) is found such that $\sigma(\alpha^i) = 0$. In such a case α^i = error's root and σ(x) = error's position. After that the Number of zeroe for the error-finder-polynomial σ(x) is computed and a comparison made between it and the polynomial degree. If a match is found, the error vector is updated and σ(x) is evaluated in all symbol positions of the code word. A mismatch indicates the presence of more errors than can be corrected. The error locator polynomial has roots from the reciprocal of the locations where the errors occurred. By solving for a minimum degree solution, the error locator polynomial is obtained.

Having calculates the coefficient values, $\wedge_{1\ldots}\wedge_v$ the error-finder-poly., now we can calculate its roots. If the polynomial is written in the form:

$$\Lambda(x) = X_1(x + X_1^{-1})X_2(x + X_2^{-1}). \qquad (13)$$

Then the value of the function will be zero if $X = X_1^1$, X_1^2…. etc.; where $X = \alpha^{-e1}, \alpha^{-e2}$. The roots, and hence the values of $X_1 \ldots X_v$, are found by trial and error, in which all the possible values of the roots (the field values $\alpha^i, 0 \le i \le n-1$) are substituted into the equation, and the results evaluated. If the expression reduces to zero, then that value of x is a root and specified the error position.

Since the first symbol of the code word corresponds to the x^{n-1} term, the search begins with the value $\alpha^{-(n-1)} = \alpha^1$, then $\alpha^{-(n-2)} = \alpha^2$ and continues to α^0, that corresponds to the last symbol of the code word.

D. Forney Algorithm

There are two approaches for computing the error magnitude; either in the frequency-domain by transforming decoding process or time domain by the Forney algorithm method. In spite that the Chien Search is no more needed in the frequency domain approach, but still the Forney algorithm is the most favorable approach, according to the simplicity of its circuit design [4]. At this moment, we know the error's position, but we do not know its values. Therefore, one's next step is to figure out its value depending on both the syndromes and error polynomial and by using Forney algorithm; which is the effective method for computing an inverted matrix. It has two phases. The 1st stage is the evaluation of the error evaluator polynomial Ω(x) at each null location by the convolution between the error polyn-omial and syndromes σ(X) (from the BMA result). The 2nd stage is the division between Ω (x) the derivative of σ (X). The result provides the error/congruent-position. If there is bit=1 inside the error symbol, then it is an error in the equivalent bit of the received symbol, and the inversion process should be made in such a case [7]. The Forney algorithm corrects the error symbol after detecting it, by constructing the error evaluation polynomial Ω(x) as expressed in the equation below:

$$\Omega(x) \cong S(X)\sigma(X) \bmod X^{2t} \qquad (14)$$

Where $S(X)= S_1 + S_2X + S_3X^2 + \ldots +S_{2t}X^{2t-1}$ is syndrome polynomial, and $\sigma(x) = (1-\beta_1X)(1-\beta_2X)\ldots= \sigma_0 + \sigma_1X + \sigma_2X^2 + \ldots + \sigma_vX^v$, (15)

is the error location polynomial. We call the first equation as the key equation. The modulo X^{2t} operation in the equation is to remove all terms of order ≥2t.

Forney's algorithm computes the value of the error as shown in the equation below:

$$e_{ji} = \frac{\Omega(X)}{\sigma'(X)}\Big|X=\beta_i^{-1} \quad (i=1, 2, \ldots, v) \qquad (16)$$

Where σ'(x) is the 1st differential of σ(X). The calculation of it can be done as below:

$$\sigma'(X) \cong \frac{d\sigma(X)}{dX} = \frac{d(\sigma_0+\sigma_1X+\sigma_2X^2+\cdots+\sigma_vX^{v-1})}{dX}$$

$$=\sigma_1 + 2\sigma_2X+3\sigma_3X^2+\ldots+(v-1)\sigma_vX^{v-1} \qquad (17)$$

We finally can form e(x) and correct r(x) just by adding (with the XOR operation) these two polynomials together [4].

VI. SIMULATION AND RESULTS

Our main aim for simulating the RS code in MATLAB is to understand the phenomenon as how the signal is being encoded and what would happen if the signal has some error and up to what extent the decoder can detect and correct errors, the performance of Reed-Solomon FEC scheme in ATM networks has been investigated with different types of signals such as a random integer signal, message signal, and speech signal.

In order to verify the capability of RS codes in detecting and correcting errors, the channel coding has been applied to control the redundancy in the transmitted information, which increased the transmission reliability and reduced the transmission resources. The uniformly distributed random integer signal was added as a medium noise to all types of signals.

There are three steps for error detection and correction process:

1) The encoder acquires the input data and creates a convolved data.
2) Some errors are introduced in the data as a medium noise.
3) Finally, the decoder function that includes Berlekamp- Massey can search, and Forney algorithm functions perform the decoding and correction process.

A. RS Encoder and Decoder for Random Data Signal

A random symbol of integers was taken as input, these random symbols were then encoded using the RS encoder, following this signal was passed through channel, these symbols were then received at the decoder end.

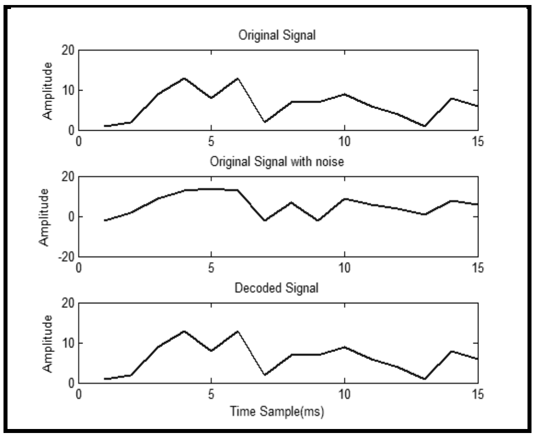
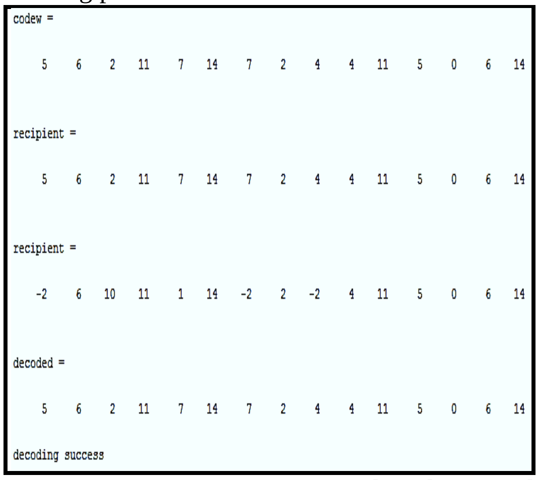


Fig. 6.Original signal, noisy input signal, and decoded signal

We see that the data is recovered according to the similarity between the input and output data, that means the decoder removed the parity bits which were added by the encoder and corrected t (t = (n-k) /2) number of errors. Here, the input data (Random signal) has been given and the decoding process was successful as illustrated in Fig. 7.



```
codew =

    5     6     2    11     7    14     7     2     4     4    11     5     0     6    14


recipient =

    5     6     2    11     7    14     7     2     4     4    11     5     0     6    14


recipient =

   -2     6    10    11     1    14    -2     2    -2     4    11     5     0     6    14


decoded =

    5     6     2    11     7    14     7     2     4     4    11     5     0     6    14

decoding success
```

Fig. 7 Data in MATLAB using random data signal

B. RS Encoder and Decoder for Message Signal

The input data symbols (Message signal) have been provided which includes three tones at 500, 600, and 700 Hz with varying amplitude for the RS encoder. In this case, one set of RS code has been used for the simulation. Figure 9 shows one cell from the transmitted message signal, since we are using an ATM network that breaks down the whole transmitted message shown in Fig. 8 into fixed size cells.

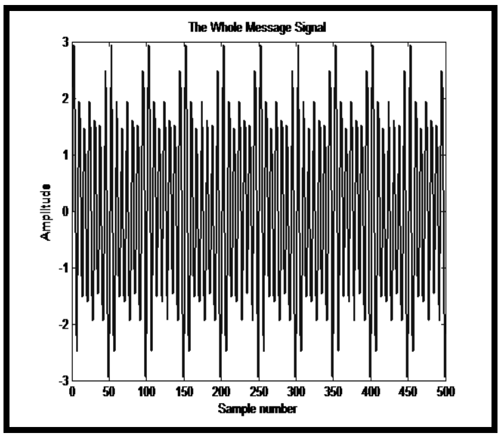


Fig. 8. Transmitted message signal

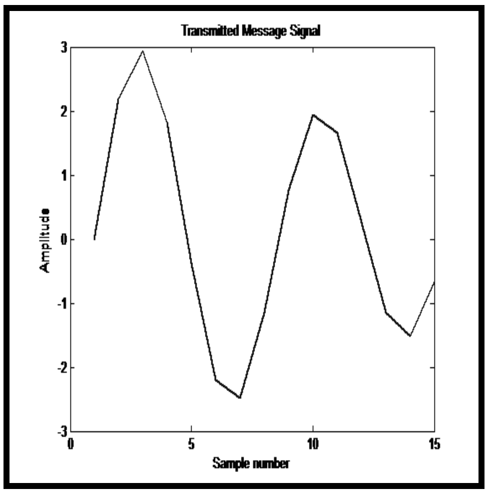


Fig. 9. One cell from the full message signal

Figure 10 shows how this cell is affected by a noisy channel, while the entire decoded message signal is shown in Fig. 11, and its decoded recovered signal is illustrated in Fig. 12.

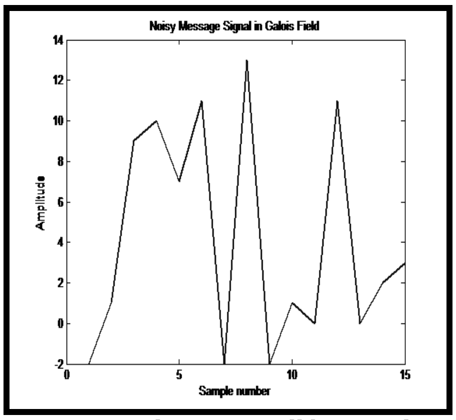


Fig. 10. One corrupted message cell by a random noise

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 5, May 2014

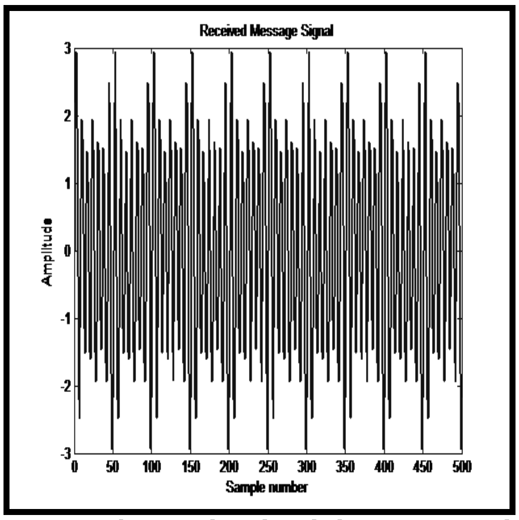


Fig. 11. The complete decoded message signal

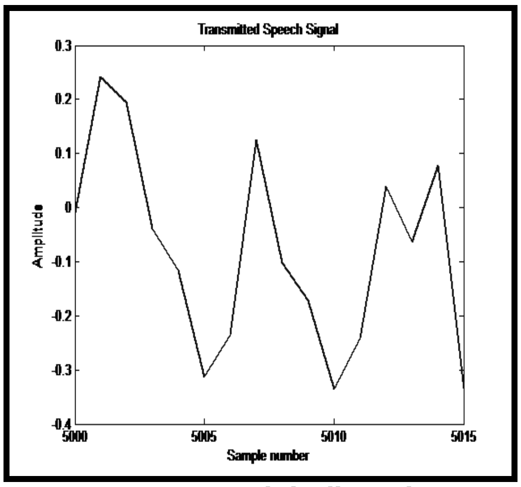
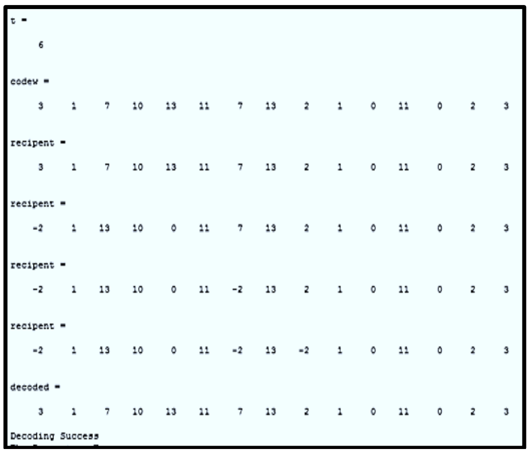


Fig. 12. Decoded cell signal

The input data (Message signal) has been given as shown in Fig. 13. We see that the decoding process was successful.



```
t =

    6

codew =

    3    1    7   10   13   11    7   13    2    1    0   11    0    2    3

recipent =

    3    1    7   10   13   11    7   13    2    1    0   11    0    2    3

recipent =

   -2    1   13   10    0   11    7   13    2    1    0   11    0    2    3

recipent =

   -2    1   13   10    0   11   -2   13    2    1    0   11    0    2    3

recipent =

   -2    1   13   10    0   11   -2   13   -2    1    0   11    0    2    3

decoded =

    3    1    7   10   13   11    7   13    2    1    0   11    0    2    3

Decoding Success
```

Fig. 13. Data using MESSAGE signal

C. *RS Encoder and Decoder for Speech Signal*

In this case, one set of RS code has been used for the simulation. Fig. 14 shows the full transmitted speech signal, while Fig. 15 shows one cell from the transmitted speech signal.

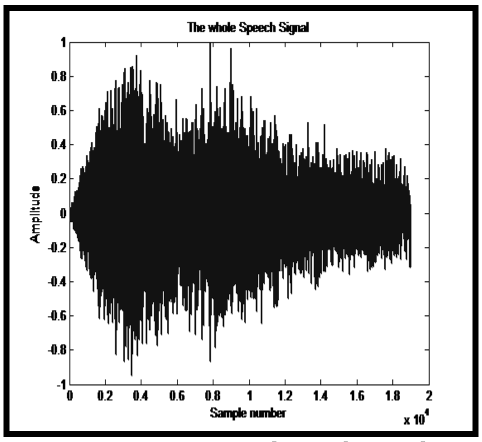


Fig. 14. Transmitted speech signal

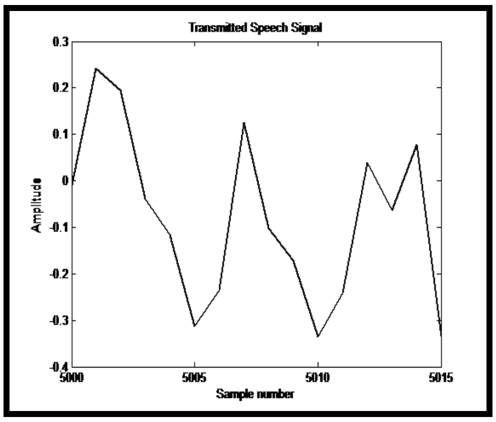


Fig. 15. One cell from the whole speech signal

Figure 16 shows how this cell is affected by the noise signal. The full decoded speech signal is shown in Fig. 17. The decoded recovered signal is illustrated in Fig. 18.

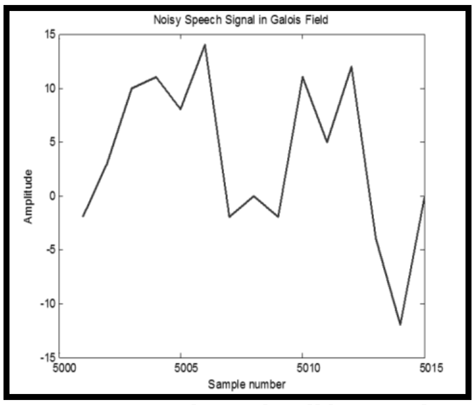


Fig. 16. One corrupted speech cell by a random noise

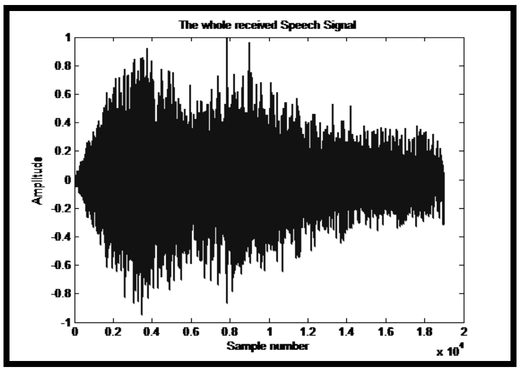


Fig. 17. The full decoded speech signal

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

TABLE I
Codeword Symbol, RS Code and Rate

Codeword Symbol	RS Code Rate	Code Rate (Rc)
15	RS (15,11)	0.738
31	RS (31,23)	0.742
63	RS (63,47)	0.746
127	RS (127,95)	0.748

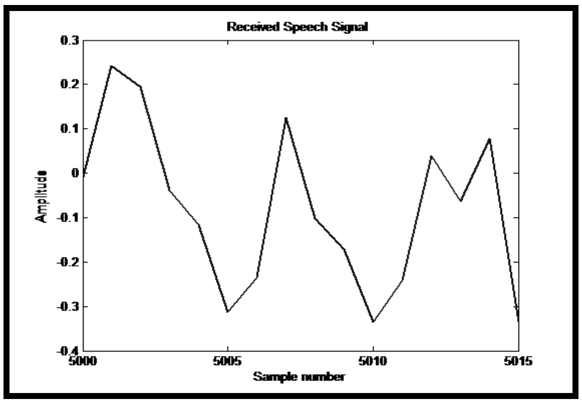
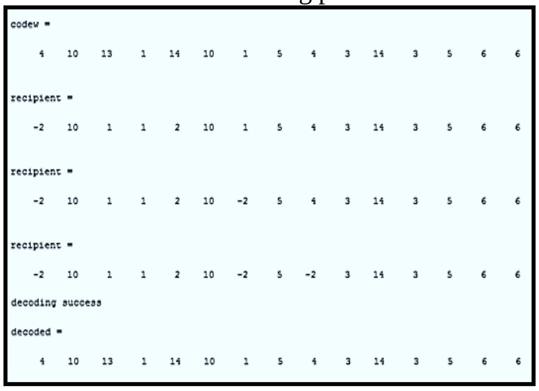


Fig. 18. Decoded cell signal

The input data (Speech Signal) has been given as shown in Fig. 19. We see that the decoding process was successful.



```
codew =

     4    10    13     1    14    10     1     5     4     3    14     3     5     6     6


recipient =

    -2    10     1     1     2    10     1     5     4     3    14     3     5     6     6


recipient =

    -2    10     1     1     2    10    -2     5     4     3    14     3     5     6     6


recipient =

    -2    10     1     1     2    10    -2     5    -2     3    14     3     5     6     6

decoding success

decoded =

     4    10    13     1    14    10     1     5     4     3    14     3     5     6     6
```

Fig. 19. Data using Speech signal

D. Code-Length Function's Performance at Steady-Code

The RS-BER Performance has been appraised upon the complete RS coded communication system over a random noisy track. The assessment of RS-Performance can be made by Code-Rate, Redundancy and Error-Correction strength; since it is one of the Block-Code families. The utilized RS-codes groups for the simulated-communication paradigm are as follows: RS (127, 95), RS (63, 47) and RS (15, 11),in the random noise track. These groups have various code-word-symbols(n) and data-symbols (k) with nearly steady-code-rate(R_c).The used RS code rates are shown in the Table 1 below which is obtained by the division between data symbols and code word symbols (R = k/n). The code rate improved with larger data block lengths. By improving the code rate, the chance of decoding error increases because of the expected undetectable errors.

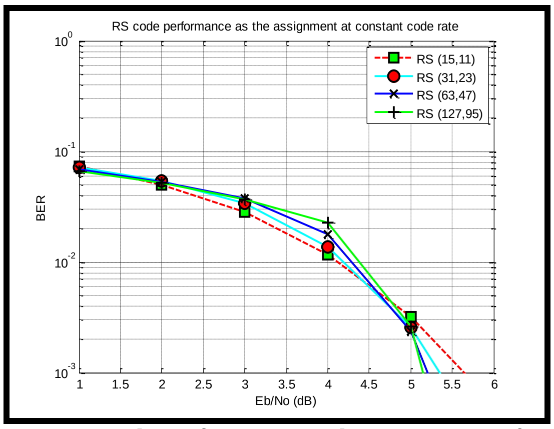


Fig. 20 .RS code performance as the assignment of code size at constant code rate

The code rate value (i.e., 0.74*)* remains constant, while the code size rises from *n*=15 symbols (4-bits/paradigm) to 127 (7-bits/paradigm). In the above Fig. 20, E_b/N_0 dB = information bit-Energy/Noise-power-density ratio and y-axis = bit error rate (BER).

The RS error-correction strength becomes further active with code-size's increment according to the above graph; because of the noise-influence reduction/large-code size. RS codes are most preferable for long block size. That is, having a higher performance by optimum RS code-word (n).

E. Performance with Same Error Correcting Capability

The empirical model has been applied to the simulation using a random noisy channel with the same error correcting capability to the next RS-codes groups respectively RS (15, 11), RS (31, 27), RS (63,59), RS (127,123).

RS-code-word (n) and data (k) symbols are increasing with maintaining a steady error-correcting strength (t=2). The BER-performance increases as a code-word-symbols (n) increase as shown in the performance curve in Fig.21, since the increment of code-length begins from n = 15 till n = 127.

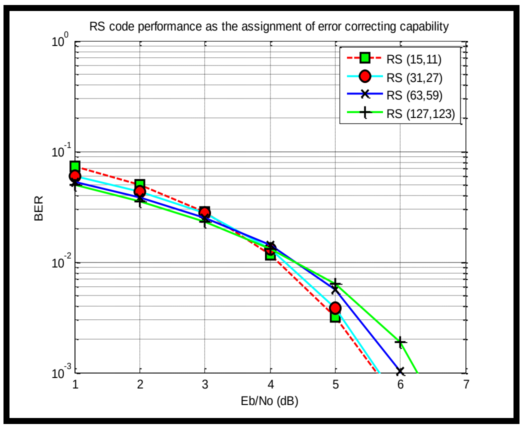


Fig. 21. Performance Evaluation as the Function of Code length

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 5, May 2014

A larger RS code means a better BER performance as it is obvious from the above figure, with constant error correcting capability (t=2) for both.

F. Performance as the Redundancy's Function

Figure 22 shows the applied simulation-model for RS (127, 125), RS (127, 123), RS (127, 111) and RS (127, 105).

The offered RS-codes groups have congruous code-word symbols (n=127) as No. of data-symbols (k) decreases from 125 to 105. Also, the code-rate (R_c=k/n) has been decreased from 0.98 to 0.83.

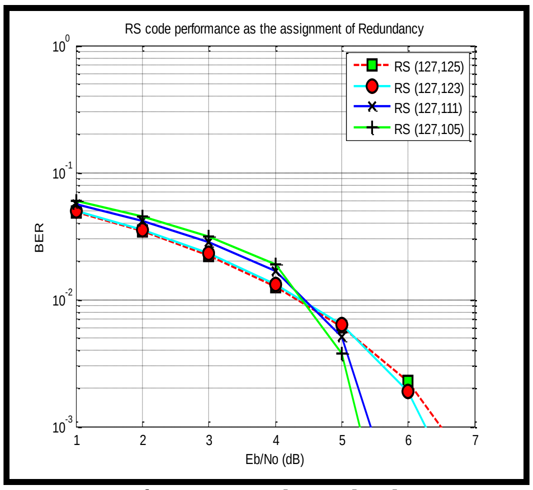


Fig. 22. RS-Performance as the Redundancy's Function

Obviously, RS-coding with Elevated code-rate (0.98) is better than low-code-rate (0.83). Regarding Fig.22, when the redundancy has increased from 2 to 22 (that means lowering code rate) the signal to noise ratio performance has also improved.

VII. HAMMING AND REED SOLOMON COMPARISION

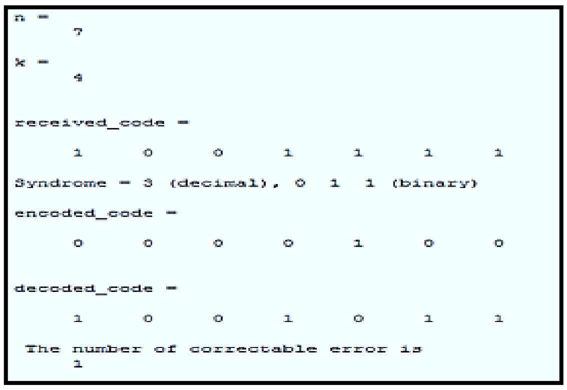
The Hamming and Reed Solomon code algorithms are compared, and their differences are briefly discussed.

- The Hamming algorithm is straightforward and can achieve as SW/HW easily. There is a Hamming-mechanism' restriction because of the restricted error-correction's strength. Hamming-code' ability corrects 1-bit errors and detects only 2-bits errors. A Hamming code is usually defined as (2^n-1, 2^n-n-1) where:-

n	$\Longrightarrow$	The number of bits
2^n-1	$\Longrightarrow$	The block size
2^n-n-1	$\Longrightarrow$	The number of data bits in the block.

Prevalent Hamming-code' sizes are (7, 4), (15, 11) and (31, 26). Since 7-bit-block contains (4-bits data + correct-code). The same thing is considered for (15, 11) and (31, 26) codes. Therefore, the Hamming algorithm for Error Correction Code (ECC) computation should be used with particular consideration.

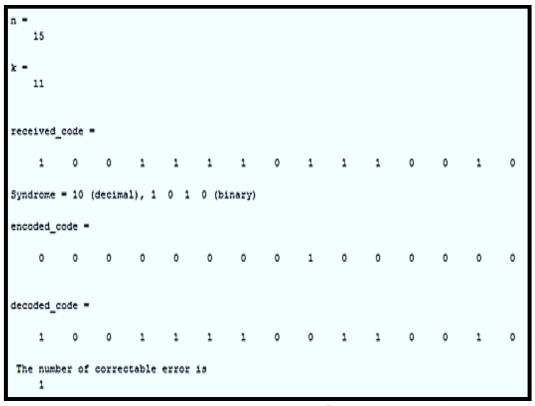
After applying the hamming code (7, 4) which is considered as a famous single error correcting code, we see that the maximum number of correcting errors is 1 as shown in the Fig. 23, while in the Reed Solomon code it is more than one error.



```
n =
       7

k =
       4


received_code =

       1     0     0     1     1     1     1

Syndrome = 3 (decimal), 0   1   1 (binary)

encoded_code =

       0     0     0     0     1     0     0


decoded_code =

       1     0     0     1     0     1     1

 The number of correctable error is
       1
```

Fig. 23. Hamming code (7,4)

The Fig. 24 shows that the Hamming code (15, 11) has also detected only two bit errors and corrected one of them.



```
n =
    15

k =
    11


received_code =

    1   0   0   1   1   1   1   0   1   1   1   0   0   1   0

Syndrome = 10 (decimal), 1  0  1  0 (binary)

encoded_code =

    0   0   0   0   0   0   0   0   1   0   0   0   0   0   0


decoded_code =

    1   0   0   1   1   1   1   0   0   1   1   0   0   1   0

 The number of correctable error is
    1
```

Fig. 24. Hamming code (15, 11)

Even after increasing the values of code word and data symbols, the correcting ability of the Hamming code (31, 26) remains the same as before (detecting two errors and correcting one). But when we take the same value of Hamming code word (15, 11) and implementing it in RS code, we get 4 corrected errors, which is more than what we get at Hamming code.

TABLE III
Code word Symbol, Hamming-Set and R_c

Codeword Symbol	Hamming Code Set	Code Rate (R_c)
15	Hamming (15,11)	0.738
31	Hamming (31,23)	0.742

Hamming (15, 11) and Hamming (31, 23) are the Hamming-Code groups utilized in the communication model; in order to compare their results with RS codes. They have diversified code-word paradigms (n) and data paradigms (k) with steady code-rate. The code-rate's value remains steady= (0.74) while its code size increases from n=15 paradigms (4-bits/paradigm) to 31 (5-bits/paradigm). Clearly, regarding Fig. 25 the increment in Hamming code-lengths made its codes more better; due to noise impact's reduction/larger code-length; while Reed Solomon codes with the same code length, we see that the noise effect is reduced more than the Hamming codes.

As we see when we used Hamming (15, 11), the SNR is 5.856 dB, but when using the same code length in Reed – Solomon RS (15, 11), the SNR became 5.522 dB. Also, the improvement in SNR is equal to 5.706 for Hamming (31, 23) while the improvement in SNR is equal to 5.234 dB for RS (31, 23). That means RS codes are extremely efficient against burst errors and offer significant improvements in the SNR needed to achieve a given BER.

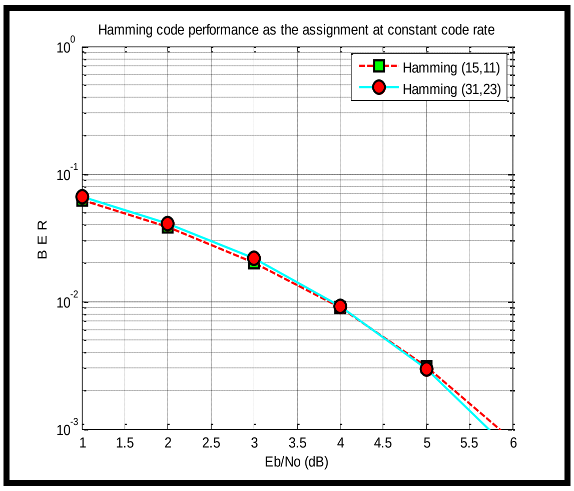


Fig. 25. Hamming code performance as the assignment at constant code rate

While Fig.26 shows a Hamming-Code with rising R_c = (0.83) is same as decreased R_c= (0.65); but in RS coding, the higher R_c= (0.98) is better than (0.83).

Referring to the curve, when redundancy increased (lowering code rate), the signal to noise ratio has not improved but in RS codes the signal to noise ratio has improved as the redundancy increased.

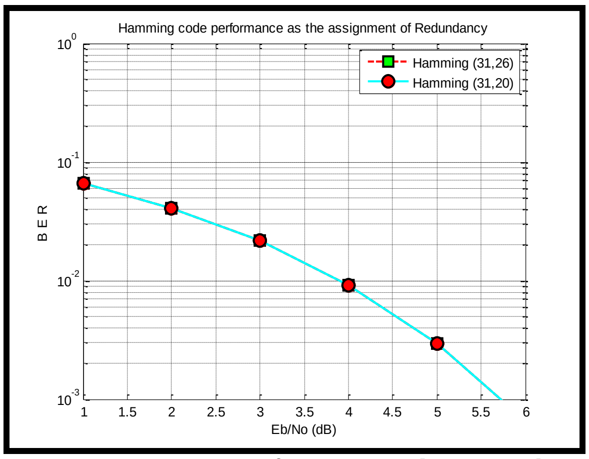


Fig. 26. Hamming Performance Evaluation as the Function of Redundancy

VIII. CONCLUSION

The important characteristic of RS (FEC) error detecting and correcting coding technique for ATM technology has been discussed. The permitted real maximum code rate depends on the used code of correction error. The actual information appears unaltered in the encoded Hamming Performance Evaluation as the Function of Redundancy data, due to the fact that RS is a systematic code. Some redundant bits are added for detection and correction of errors. FEC codes are considered as a powerful tool in combating transmission errors caused by a noisy channel and it allows communications to achieve the same level of transmission reliability. It is quantified by the Bit-Error-Rate, which potentially give a large benefit because it reduces the amount of power needed for the transmission of data. In the encoder, some redundant symbols were added using a generator polynomial which is also used for error value and position at the decoder. In RS encoding, the information remained the same, but an extra parity symbol has been added. Since, before the transmission of data, the encoder has attached a parity symbol using a predefined algorithm (encoding). The decoder first corrects the symbols, then removes the redundant parity symbols from the code word and produces the recovered input data. From the simulation result, the decoder corrected more than 6 numbers of errors. The RS-schema improved BER-performance; since the RS-performance has been examined through diversified criteria (Code-Size, Redundancy, R_c). The simulation results clearly showed that the BER performance curve improved as the code size (n) increased at a constant correcting capability. The BER-Activity improved with the raised Redundancy (from 2-22 paradigms). The Reed-Solomon coding that we have used for simulating the detection and correction in MATLAB present an adequate performance for application in ATM network. The given-results were advantageous in identifying the Simulation's results. Reed Solomon will persist in perfecting the communication performance.

REFERENCES

[1] HARRY G. (2005). Connection -Oriented Networks, SONET/SDH, ATM, MPLS and OPTICAL NETWORKS. John Wiley & Sons Ltd.
[2] HONGYUAN CHEN, TAKAHASHI K., ERKE T. (1996). The Analysis of the Application of ARQ Mechanism in ATM Network, IEEE Conference Proceeding on International Conference on Communication Technology Proceedings, ICCT'96. **1**, 263-266.
[3] MARTYN XRILEY, IAIN Richardson. (2001). An introduction to Reed Solomon codes: principles, architecture & implementation, prentice-hall.
[4] SYED MOHD. (2005). Evaluation of RS Coding performance for Mary modulation in AWGN and multipath fading channels. Project Report, University Technology Malaysia.
[5] C.K.P. CLARKE, "Reed-Solomon Error Correction", Research and development, British Broadcasting Corporation, R and D white paper WHP 031 July 2002.
[6] R. MICHELONI, A. MARELLI and R. RAVASIO, "Error Correction Codes for Non-Volatile Memories", Library of Congress control number 2008927177, 2008 Springer Science + Business Media B.V.
[7] HARRY G. (2005). Connection -Oriented Networks, SONET/SDH, ATM, MPLS and OPTICAL NETWORKS. John Wiley & Sons Ltd.