01 Jan 2007

Improvement of CAN BUS Performance by using Error-Correction Codes

Krishna Chaitanya Emani

Maciej Jan Zawodniok
Missouri University of Science and Technology, mjzx9c@mst.edu

Y. Rosa Zheng
Missouri University of Science and Technology, zhengyr@mst.edu

Jagannathan Sarangapani
Missouri University of Science and Technology, sarangap@mst.edu

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/ele_comeng_facwork/1082

Recommended Citation

K. C. Emani et al., "Improvement of CAN BUS Performance by using Error-Correction Codes," *Proceedings of the IEEE Region 5 Technical Conference, 2007*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2007.
The definitive version is available at https://doi.org/10.1109/TPSD.2007.4380382

IMPROVEMENT OF CAN BUS PERFORMANCE BY USING ERROR-CORRECTION CODES

Krishna Chaitanya Emani, Keong Kam, Maciej Zawodniok
Yahong Rosa Zheng, Jagannathan Sarangapani
Department of Electrical and Computer Engineering
University of Missouri-Rolla
1870 Miner Circle, Rolla, MO 65409-0040
Email: {ke3tc, kwkx8d, mjzx9c, zhengyr, sarangap}@umr.edu

***Abstract*— In this paper, two variants of the Hybrid Automatic Repeat Request (HARQ) scheme for CAN bus are presented. The basic HARQ uses error-correction code based on the Reed-Solomon (RS) technique and the Cyclic Redundancy Check (CRC) method to detect errors. The second scheme uses the cyclic error-correction method instead of the CRC error-detection method to further improve the throughput. Moreover, the second scheme uses no additional bit overhead when compared with the basic HARQ scheme. This paper presents the performance of the proposed schemes using MATLAB and NS2 simulations. Experimental data of error patterns were used for realistic evaluation.**

The basic HARQ method corrects 100% of error bursts shorter than 7 bits. When the burst length falls between 7 to 10 the scheme corrects between 86% and 56% of the corrupted frames. Network Simulator (NS2) simulations showed that the throughput increased by 92% when the user message size was increased from the standard 64 bits to 512 bits as a result of reduced overhead per user bit.

I. Introduction

CAN has become a de facto standard for industrial networks. It is a serial communication protocol that supports distributed real-time control systems with a high level of fidelity [1]. However, the communication speed is reduced to improve the robustness. At present CAN is being used for several applications on vehicles such as data transfer between sensors and actuators on an automobile [2]. A recent trend in the automotive industry has increased the complexity of these types of applications. As a result, the number of nodes connected to the CAN bus has increased. Additionally, safety and efficiency requirements such as latency, high data rate, immunity to noise, and error-detection capability are challenging the current CAN capabilities.

Several factors reduce the efficiency of the CAN bus. The main factors are as follows: errors caused by electro-magnetic interference (EMI) and inefficient stop-and-wait retransmission, high bit overhead, a broadcast system that requires retransmission when any node raises an error flag, and also the length of the CAN bus. Current methods, which include Cyclic Redundancy Check (CRC), Bit Monitoring, and Bit Stuffing [4], can only detect an error. When an error is detected, the receiver sends an Automatic repeat ReQuest (ARQ) to the transmitter asking for retransmission [1]. These methods enable error-free data transfer at the expense of the throughput. For high-speed CAN the efficiency is about 30%. In contrast, the proposed scheme aims to reduce the retransmissions by correcting the errors caused by EMI and Additive White Gaussian (AWGN) noises, thus leading to improved bandwidth utilization. Typically the frame contains only a few bit errors, which can easily be mended by an error-correction code. Consequently, the retransmission is required only when the frame is highly corrupted with 10 to 15 error bits.

In this paper, the new method of using HARQ is proposed to improve the performance of the CAN bus. The user bits and the CRC bits are additionally protected by an error-correction code, which increases the probability of a successful transmission. The receiver decodes the corrupted frame with the error-correction decoder and then checks for errors. If the error still exists then the receiver asks for a retransmission [6]. Hybrid Automatic Repeat Request (HARQ) has become an integral part of many packet communication systems. Results from experiments conducted in laboratory at University of Missouri-Rolla by Fei Ren show that EMI is the only source of errors that affects the throughput. Typically, EMI corrupts a burst of bits in the frame. The proposed method uses RS codes, which are suitable in combating the burst errors. Another technique proposed in this paper is the use of cyclic codes instead of CRC to correct the remaining single bit errors. Comparison is made between the performances of the two methods to check for the gain that cyclic codes offer.

The rest of this paper is organized in the following manner. In Section II, the proposed scheme is explained in detail. In Section III, the implementation guidelines are provided. Next the simulation results of MATLAB and NS2 are discussed in Section IV. Section V indicates the computational complexity of using R-S codes in the CAN bus. Conclusions are given in Section VI. Proposed future work and further studies are mentioned in Section VII.

II. Proposed Scheme

In this paper the proposed methods are:

1. HARQ scheme which uses R-S codes and Cyclic Redundancy Check (CRC).
2. Modified HARQ scheme which uses R-S and Cyclic codes instead of CRC.
3. Redesign of frame to reduce frame's bit overhead.

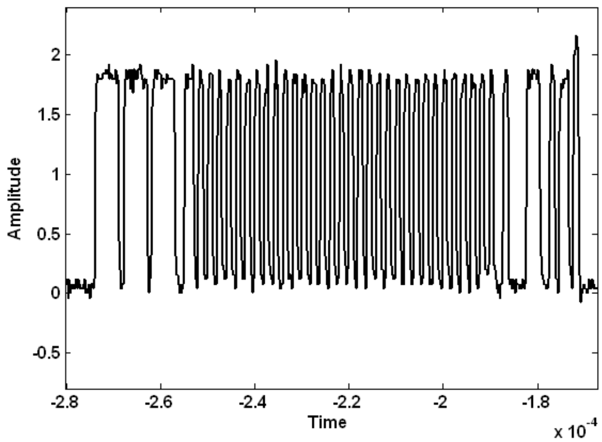

Fig. 1. CAN Frame without EMI

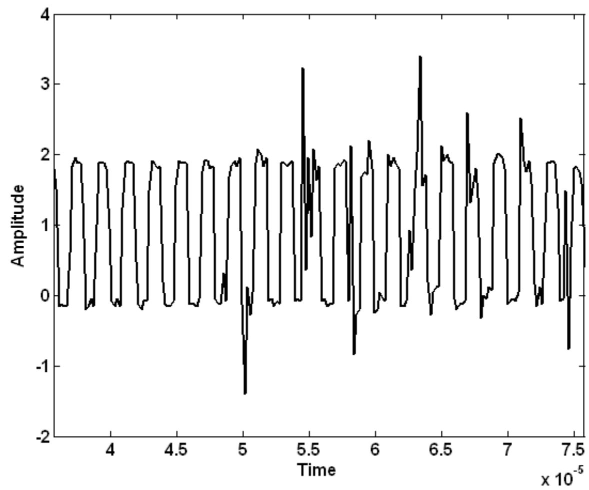

Fig. 2. CAN Frame with EMI

The first method uses RS codes along with ARQ which is being used by the present CAN. Minor modifications are needed for the CAN frame which will be backward compatible with the existing CAN. The second method replaces the ARQ bits with cyclic error-correction bits. The third idea presented in this paper discusses the need for increasing the user message length to reduce the effect of frame header overhead.

A. Introduction to the proposed Reed-Solomon Codes

First the effect of Electro-Magnetic Interference (EMI) has to be analyzed to design a suitable HARQ scheme. The experimental tests conducted by Fei Ren at a laboratory in University of Missouri-Rolla, provided a valuable input about the EMI and its effect on the CAN data frame. Figs. 1 and 2 show the CAN frame without EMI, and with EMI respectively. The spikes shown in Fig. 2 are the result of EMI acting on the CAN bus. Careful observation of the signal shows that EMI adversely affects only about 3-5 bits of the CAN data frame. CAN bus receiver or any node on the CAN bus will discard frame even with single bit error. However, with error-correction schemes these errors can be corrected.

In general error-correction coding is the technique used to correct the errors in the received data. It introduces systematic redundancy in the transmitting data in order to combat with the error. Block codes are popular category in the error-correction schemes. Block codes are defined by (n, k) where k is the number of input data bits and n is the number of bits in the encoded frame. These codes are simple to implement and have low computational complexity than the other competing coding schemes. This paper proposes R-S type block code. R-S codes are suited to combat burst errors. Fig. 2 shows that the errors that are induced are due to EMI are of burst type. Hence, these errors can be easily corrected by the proposed scheme. Reed-Solomon (R-S) codes are also a kind of block codes. (R-S) codes are non-binary cyclic codes with symbols made up of m-bit sequences, where m is any positive integer having a value greater than 2 [5]. R-S (n, k) codes for m-bit symbols exists for all values of n and k satisfying the condition

$$0 < k < n < 2^m + 2 \quad (1)$$

where k is the number of symbols being encoded and n is the number of symbols in the encoded block. The values of n and k can be selected according to the requirement. The relation between the values of (n, k) can be given as

$$(n,k) = (2^m - 1, 2^m - 1 - 2t) \quad (2)$$

where t determines the number of symbols that the code can correct. Encoded block size n is determined by the expression given in Equation (2) and k can be any value less than n with the only condition that n-k should be even. The value of t is given by the equation

$$t = \frac{n-k}{2} \quad (3)$$

Depending on the error-correction performance needed, the values of (n, k) can be selected and the number of symbols that can be corrected will always be t irrespective of the number of bits corrupted in each symbol. This is one of the major advantages of using R-S codes. They can be used to counter the burst errors in CAN bus that have an average length of 3 to 5 bits.

B. Description of the CAN Frame Structure

First, the current CAN data frame structure is presented and then the design of the algorithm is explained. Two versions of CAN frames are currently in use: the standard version and the extended version [1], shown in Fig. 3 and Fig. 4. The number of bits in each of the fields for both versions is mentioned below.

Standard Frame Structure: Header : 19 bits; Data field: 64 bits; CRC field: 15 bits; Tail:10 bits.

Extended Frame Structure: Header: 39 bits; Data field: 64 bits; CRC field: 15 bits; Tail:10 bits.

SOF	Arbitration	Control	Data Field 64 bits	CRC	Ack	EOF
1 Bit	12 bits	6 bits		15 bits	2 bits	7 bits

Fig. 3. Standard CAN Frame with Individual Fields Shown

SOF	Arbitration	Control	Data Field 64 bits	CRC	Ack	EOF
1 Bit	30 bits	6 bits		15 bits	2 bits	7 bits

Fig. 4. Extended CAN Frame with Individual Fields Shown

Two optional bits are present in the header of both versions of CAN frames: the Reserved bit (R0) and Identifier Extension bit (IDE) in the standard frame, and the Substitute Remote Request (SRR) and Identifier Extension bit(IDE) in the extended frame structure. An additional bit is in the tail, which is common for both versions and is called the CRC delimiter [3]. In the current version of CAN the CRC is used to detect errors. In the proposed scheme the CRC method is supplemented by R-S encoder. The proposed scheme do not require any significant change in the CAN frame structure. The frame is appended by error-correction bits at the end. Fig. 5 shows the extended CAN frame encoded with R-S scheme. Reserve bits are present in the CAN frame to indicate the presence of the R-S field to make it compatible with the existing CAN.

III. Implementation

Fig. 6 shows the algorithm used to implement R-S codes on the CAN bus. The arrows and the blocks shown with solid lines is the flow for R-S codes with CRC. The blocks shown by dashed lines represent the second method of replacing the CRC bits with cyclic codes. Random data is generated and each time 64 bits are selected and a header is added to the frame. Encoded CRC bits are appended, and the tail is added to make the frame 128 bits. This frame is then encoded using the R-S encoder. The parameters for the R-S encoder are

$$m = 5; n = 2^m - 1 \Rightarrow n = 31; k = 27 \Rightarrow t = \frac{n-k}{2} \Rightarrow t = 2 \quad (4)$$

The CAN frame length is equal to 128 bits. Hence a coding scheme that will encode all 128 bits is needed. The value chosen of m=5 gives the value of n=31, and choosing k=27 enables the encoding of all the bits of the CAN frame. Zeros are appended to the CAN frame to make the message length equal to 135 bits. Once the frame is encoded the appended zeros are removed and the CAN frame is transmitted along with the error-correction bits.

Then, the frame length increases from 128 to 148 bits with 20 additional error-correction bits. The frame is then corrupted with a burst error varying from 3 to 10 bits. At the receiver, the zeros are appended back to make the length equal to 155 bits and then the corrupted frame is decoded using the R-S decoder. The zeros are removed from the decoder output to obtain 128 bits, which are then checked for errors using CRC. If errors are detected then the receiver asks for a retransmission. If no errors are detected then the receiver accepts the frame. The burst length is then incremented by 1 until it is less than 10 bits.

SOF	Arbitration	Control	Data Field 64 bits	CRC	Ack	EOF	Additional RS
1 Bit	30 bits	6 bits		15 bits	2 bits	7 bits	bits (20)

Fig. 5. CAN Frame with the Additional R-S Bits Added

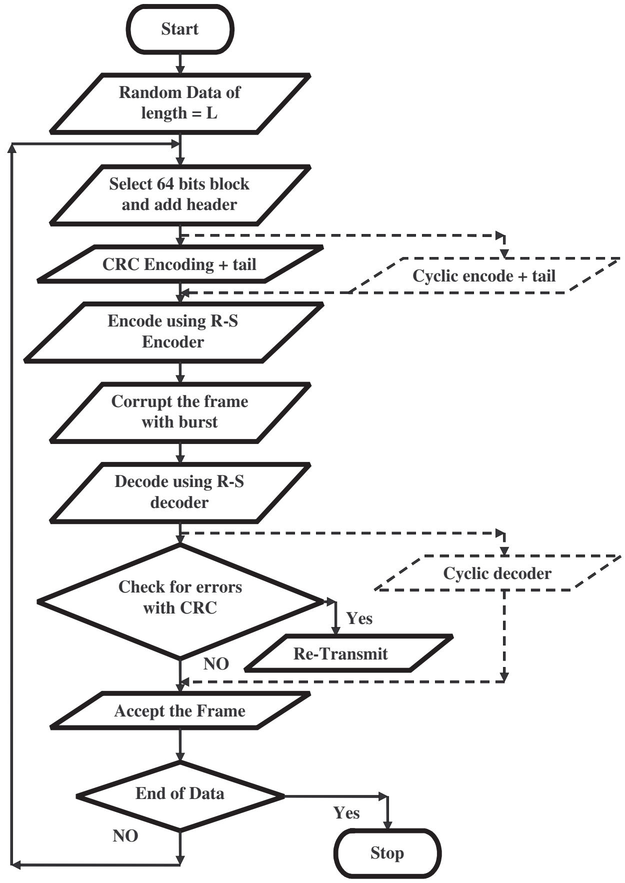

Fig. 6. Flowchart Depicting Implementation of R-S with CRC and R-S with Cyclic Codes for Randomly Generated Data and Random Burst Length

Random noise also introduces some error in to the frame. These errors are random in nature and most often are not burst errors. Some bits are corrupted randomly depending on the value of SNR. An idea to combat these random errors introduced due to SNR is to implement cyclic codes in place of CRC to correct the remaining random errors once the R-S codes have corrected the burst errors. A simulation study was made to check for any improvement. The cyclic codes used for the simulation had the following parameters: n=60, k=53. The values of n and k were chosen such that the complete CAN frame including the header would be encoded using the cyclic code.

In the flowchart shown in Fig. 6, the flow of encoding the

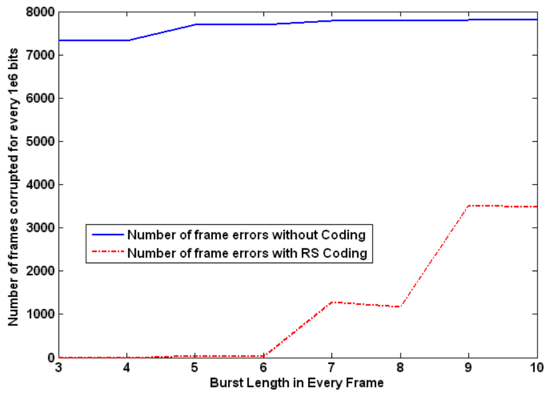

Fig. 7. Comparison of the Number of Correct Frames Received

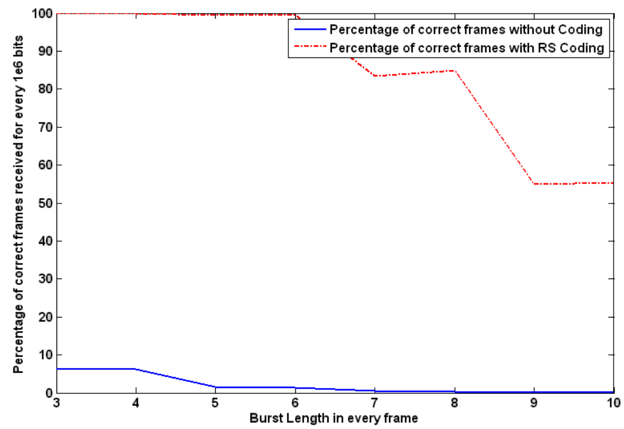

Fig. 8. Comparison of the Percentage of Correct Frames Received

data is almost the same with the only difference being the cyclic encoding and decoding. This difference is shown in the flowchart with dashed lines. The R-S encoded frame is encoded using cyclic codes and then transmitted over the CAN bus. At the receiver the R-S decoded frame is then decoded using cyclic codes to obtain the final decoded bit sequence. In this case the final bit sequence has to be accepted even if errors exists because this method has no mechanism for error detection in the final decoded frame. Comparisons have been made for different parameters to check for any improvement in the number of correctly received packets.

IV. Results and Observations

A. MATLAB Simulations and Results

Fig. 7 shows the number of error frames in the received with and without R-S codes. Fig. 8 shows the percentage of correct frames received for varying burst lengths. The SNR value for all the simulations is assumed to be 15 dB. R-S codes were able to correct all errors in the case of burst lengths smaller than or equal to 6, since a burst always fits into two R-S symbols that can be corrected. The probability of correcting errors for a burst length of 7 or higher is lower than 100%, because the burst can span over more than two R-S symbols. For example, when the burst length is equal to 7, the burst can be distributed over three symbols [1, 5, 1]. In such cases, the R-S codes cannot correct the frame. The probability of correcting errors decreases with the burst length because in more and more cases the burst corrupts three symbols. The R-S coding will be unable to correct the frame in such cases. Hence for a burst length greater than or equal to 7 bits, the percentage of correct frames received is decreasing. However, when compared with scheme without R-S coding, the performance is improved by 80%. Fig.7 shows the number of frames that are received with errors for the case of no FEC and for the case of R-S encoding.

Fig. 9 shows the graphs for comparison between the percentage of correct frames received for only R-S encoding and R-S and cyclic coding together.

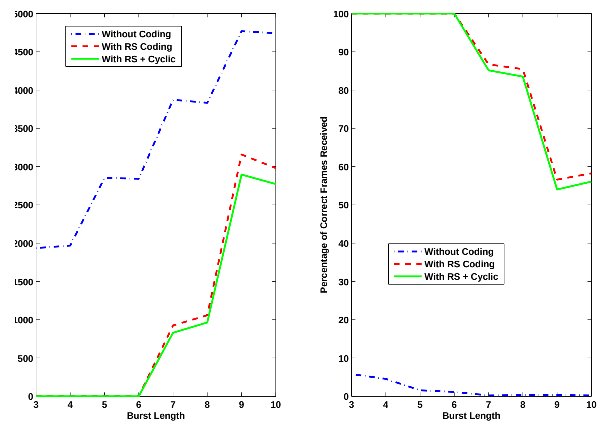

Fig. 9. Performance Comparison of Encoding Using Only R-S Codes and R-S Plus Cyclic Codes

As is evident from the Fig. 9 the improvement in the performance of R-S with cyclic codes is 200 bits at high burst length. Cyclic codes in addition to the R-S codes can only correct bit errors but are not successful in correcting the entire frame. The first sub-plot of Fig. 9 shows the number of bit errors in the three different cases. When cyclic codes and R-S codes are used together the number of bit errors are less than the case with R-S codes only. When the percentage frames corrected is observed the R-S with cyclic code does not correct more frames. Another major disadvantage with R-S plus cyclic codes method is the fact that the receiver cannot detect the errors in the received frame and can never receive the corrected frame. The simulation results show that the best way to improve the performance of CAN is to use Hybrid ARQ, which ensures correct transmission of the complete data. Moreover, AWGN does not seem to affect the CAN frame because in the CAN bus the SNR values are so large that random noise cannot play a vital role in inducing errors [10]. EMI causes most of the errors in CAN communications.

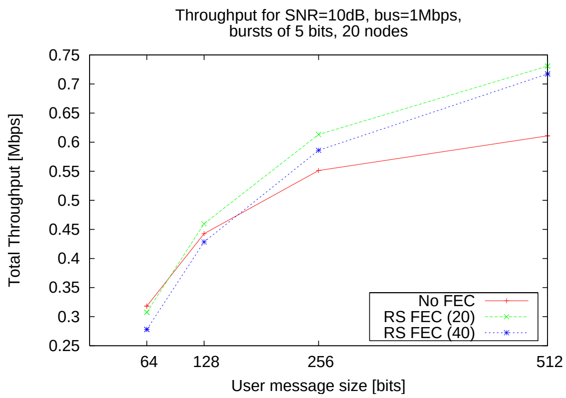

Fig. 10. Total Throughput for 20 Nodes with Size of User Message

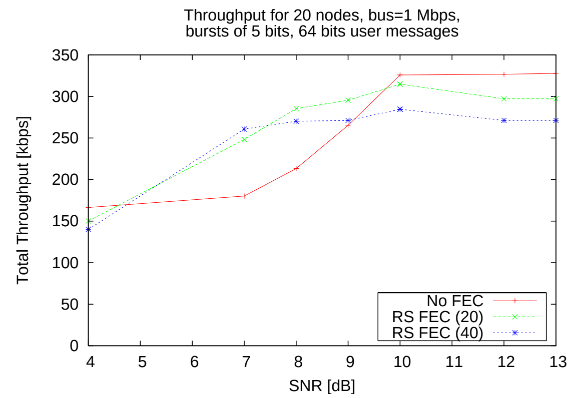

Fig. 11. Total Throughput for 20 Nodes with a 64 bit User Message

However, for higher transmission speeds the AWGN related error rate will increase thus justifying the addition of cyclic codes.

B. NS2 Simulation Results

The network simulator (NS2) has been used to simulate network of 20 nodes connected to the CAN bus. Additionally, the NS2 has been modified to simulate CAN bus operation, use files with an error sample, and perform error correction based on the Reed-Solomon (R-S) method. Two R-S variants have been simulated with 20 and 40 extra code bits, which were encoded for 5 bit symbols. The simulations were repeated with varying the user message size and SNR level.The simulations were run for network of 20 nodes, the bus running at 1Mbps, signal-to-noise ratio (SNR) equal to 10dB, and error burst of 5 bits. In NS2 simulations the SNR is varied from 4 to 13.

Fig. 10 illustrates the throughput with varying user message size for cases with R-S code with 20 and 40 extra code bits. The throughput increases with the user message size because the percentage of overhead bits in the transmitted frames decreases. Consequently, a higher percentage of the bandwidth is utilized for transmitting the user data. However, the usage of larger frames increases the probability of error occurrence in a frame. As a result, the increase of frame size does not result in proportional improvement of the throughput. In case of error-correction schemes, the throughput improves over the scheme without error-correction by up to 15%.

Fig. 11 illustrate total user throughput for varying Signal to Noise Ratio (SNR). When no error-correction scheme is used, the throughput is reduced since the errors result in retransmissions thus reducing bandwidth utilization. When error-correction codes are used the throughput increases with SNR since the error-correction codes reduce retransmissions by correcting a number of bit errors. The RS40 outperforms the RS20 code for SNR equal to 7dB since it can correct larger number of bits. When the SNR is lesser than 10dB, both RS40 and RS20 can correct all errors thus the throughput saturates. However, when the number of errors is low for high SNR, the additional overhead of RS20 and RS40 codes reduces the maximum throughput when compared to case without error correction scheme.

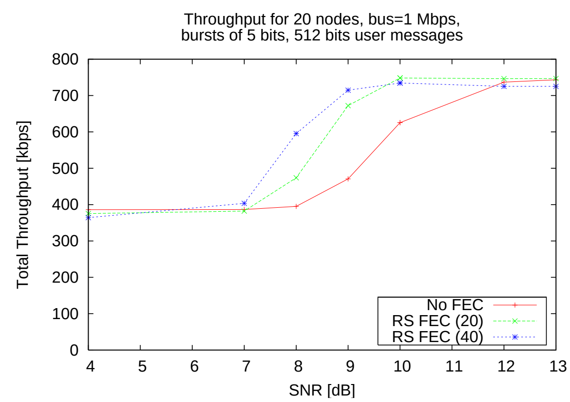

Fig. 12. Total Throughput for 20 Nodes with a 512 bit User Message

Also, when the user message size is increased to 512bits, the analogous correlation of throughput with SNR value are observed, as shown in Fig. 12.

In conclusion, an increase in the packet size will improve user the throughput for the same conditions (SNR, and the number of nodes). However, this increase is limited by an SNR value because the error probability increases with the packet size. Furthermore, the user throughput can be improved by the addition of error-correction coding. This improvement is observed for a range of SNR from 5 to 10 dB, which corresponds to the cases when the EMI pulse reduces SNR and causes errors. Hence, the performance in a noisy environment can be improvement by the usage of error-correction codes. However, the error-correction code should match the typical error pattern because the error-correction bit-overhead will reduce the bandwidth utilization.

V. Computational Complexity of Reed-Solomon Codes

Reed-Solomon codes are capable of correcting (*n*-*k*) error symbols and the computational complexity of the operation is determined by the values of *n* and *k*. High value of n increases the computational complexity of the system. Another important factor that should be considered when R-S codes are used is the fact that the R-S algorithm operates over a Galois field. Typical property of a Galois field is the closure property. Any operation (+, -, x, / etc) on a Galois field element results in an element present in the Galois field [8]. And an R-S encoder and decoder have to perform these operations in order to be used for real time applications. Special hardware and software have to be developed to implement these Galois field operations.

VI. Conclusions

1) The proposed HARQ for CAN bus improved the throughput by up to 34% over basic CAN.
2) NS2 simulations showed that for HARQ, the throughput increased due to error-correction, which overcame the additional error-correcting bits overhead.
3) NS2 simulations showed that the throughput of the system increased as the user message size increased from 64 to 512 bits.
4) The HARQ system outperformed the standard CAN when the SNR of the system was between 6 dB to 11 dB.
5) The replacement of the Cyclic Redundancy Check bits (CRC) by cyclic codes reduced the number of error bits by 10% but could not improve the performance of the the system in terms of receiving the corrected frames.

VII. Future Work

The proposed HARQ scheme checks for errors in the decoded frame and asks for retransmission if errors are detected. The transmitter retransmits the whole frame as it is without any modification. However, this is a waste of bandwidth because some additional error-correction bits transmitted to the decoder will correct more errors and might be able to decode the correct frame. This technique is called the Hybrid type-II ARQ [7]. Future studies will evaluate the effect of using Hybrid type-II ARQ for CAN bus, as well as compare the performance of the two methods in terms of the percentage of correct frames received, throughput, and computational complexity. Computational complexity of Raptor codes is less compared to the computational complexity of R-S codes [9]. Further studies will also evaluate the performance difference between Raptor codes and R-S codes in terms of computational complexity and error correcting capability.

VIII. Acknowledgments

The authors acknowledge the support and help provided by several sources and individuals. This project is funded by Caterpillar INC. under the 2006 University Challenge Program. Mr. Fei Ren, a graduate student at the University of Missouri-Rolla in the department of Electrical and Computer Engineering, provided some valuable information about the effects of EMI. Mr. Wayne Brett of Caterpillar provided technical guidance on the project.

References

[1] Robert Bosch GmbH. (1991). CAN specification Version 2.0. The Bosch IC Design Center, Reutlingen, Germany. [Online]. Available: http://www.semiconductors.bosch.de/pdf/can2spec.pdf

[2] "The Application of Controller Area Network on Vehicle," W. Xing, H. Chen, H. Ding, in *Proc. IVEC*, 1999, pp. 455 – 458.

[3] Wikipedia. Controller Area Network. [Online]. Available: http://en.wikipedia.org/wiki/Controller_Area_Network

[4] KVASER. The CAN Protocol Tour. [Online]. Available: http://www.kvaser.com/can/protocol/index.htm

[5] DIGITAL COMMUNICATIONS Fundamentals and Applications by Bernard Sklar, 2nd Edition. Pearson Education, ISBN: 81-7808-373-6.

[6] "Reliability-based type II hybrid ARQ schemes". Tripathi, V. Visotsky, E. Peterson, R. Honig, Communications Research Labs, Motorola Inc. Schaumburg, IL, USA. IEEE International Conference on Communications 2003. Volume: 4, pages: 2899- 2903.

[7] "Hybrid ARQ with Random Transmission Assignments," E. Soljanin, R. Liu, and P. Spasojevic, DIMACS Workshop on Network Information Theory, May 2003.

[8] "Raptor versus Reed Solomon forward error correction codes," Ufuk Demir, Ozlem Aktas. International Symposium, Computer Networks, 2006, pages: 264- 269.

[9] White paper - Why Digital Fountain's Raptor Code Is Better Than Reed-Solomon Erasure Codes for Streaming Applications. Digital Fountain, Inc.

[10] "A physical layer for the CAN bus using modulated PLC." David M. Davenport and Ralph T. Hoctor, International Symposium on Powerline Communications and its Applications 2005. Page(s):176 - 180