

A RECONFIGURABLE ADAPTIVE FEC SYSTEM FOR RELIABLE WIRELESS COMMUNICATIONS

Kazunori Shimizu[†], Nozomu Togawa^{††,‡}, Takeshi Ikenaga[†], Masao Yanagisawa^{††}, Satoshi Goto[†], and Tatsuo Ohtsuki^{††}

[†] Graduate School of Information, Production and Systems, Waseda University

^{††} Dept. of Computer and Science Engineering, Waseda University

^{†††} Dept. of Information and Media Sciences, The University of Kitakyushu

[‡] Advanced Research Institute for Science and Engineering, Waseda University

ABSTRACT

This paper proposes a reconfigurable adaptive FEC system. For adaptive FEC schemes, we can implement an FEC decoder which is optimal for error correction capability t by taking the number of operations into consideration. Reconfiguring the optimal FEC decoder dynamically for each t allows us to maximize the throughput of each decoder within a limited hardware resource. Our system can reduce packet dropping rate more efficiently than conventional fixed hardware systems for a reliable transport protocol.

1. INTRODUCTION

In recent years, the increasing the number of mobile internet users have spurred the research on fast data processing in wireless communications. In most of wireless environments, a rate of transmission error is higher due to attenuation, fading or interference of radio signals.

FEC (Forward Error Correction) schemes are available to correct transmission errors by adding redundancy to data packets before they are transmitted. The redundancy is used for the receiver to detect and correct errors. Clearly, The number of additional redundancies increases in proportion to the number of correctable errors at the receiver. Therefore it is an efficient approach to apply an adaptive FEC scheme to wireless channels. The adaptive FEC scheme can change an error correction capability t depending on the wireless channel condition.

Adaptive FEC schemes have been widely proposed in [2],[3],[5], and [8]. These schemes are combined with ARQ (Automatic Retransmission reQuest) to realize a reliable transport protocol. However, adaptive FEC schemes in these protocols have not been realized by dedicated hardware whose error correction capability t can be changed according to the channel condition, and they also have not taken the packet processing throughput into consideration. If packet processing throughput is improved at receivers for the reliable transport protocols, the receivers can process more packets per unit time, and can reduce packet dropping rate from a received buffer.

In high-speed communication systems, FEC CODECs are generally realized as dedicated hardware architecture to carry out high throughput. Complexity of such a dedicated hardware FEC CODEC increases depending on the error correction capability t . For example, the number of required registers is proportion to t . Consider designing a fixed hardware FEC CODEC system where an error correction capability t is four. This system can operate if t is less than four. However active area efficiency¹ decreases when this

system operates with $t < 4$. Therefore, if we apply the adaptive FEC scheme to a fixed hardware system, its active area efficiency decreases with decrease of error correction capability t .

From this point of view, we propose a reconfigurable adaptive FEC system which can maximize the active area efficiency of computational platform, and the decoding throughput also can be maximized in a limited hardware resource.

We use Reed Solomon (RS) code as an error correction code. RS decoders have been widely proposed such as in [4],[6], and [7]. Their decoders, however, do not consider that error correction capability t may be changed according to wireless channel condition, and these are based on a single fixed hardware architecture for every error correction capability t .

If a particular error correction capability t is given, we can implement an RS decoder which is optimal for t . We design the RS decoder which has 5-stage pipeline. For the optimization of the RS decoder, we focus on stage 2 which computes an error location polynomial and stage 4 which computes error magnitudes. By reconfiguring the optimal FEC decoder dynamically for each t , we can maximize the throughput of each RS decoder within a limited hardware resource. Based on this idea, our reconfigurable adaptive FEC system can reduce packet dropping rate more efficiently than conventional fixed hardware systems. We can improve data transmission throughput for a reliable transport protocol. Practical simulation results are also shown.

2. ADAPTIVE FEC SCHEME

In our reconfigurable adaptive FEC system, we use an RS code as an error correction code. Let k denote information symbols that are to be transmitted over a communication channel. These symbols are encoded into a codeword of $n (> k)$ symbols. $n - k$ redundant symbols are added to the information symbols. (n, k) RS code is guaranteed to correct up to error correction capability $t = \lfloor (n - k) / 2 \rfloor$. The number of redundant symbols increase in proportion to the number of correctable errors t . A high error correction capability t achieves high reliability, but it reduces transmission efficiency because of the additional redundant symbols. Therefore, an optimal error correction capability t has to be applied to the data which transmitted on a channel.

The adaptive FEC scheme can change error correction capability t depending on the channel condition. We utilize four RS codes which are $t = 1$, (255, 253)RS code, $t = 2$, (255, 251)RS code, $t = 3$, (255, 249)RS code, and $t = 4$, (255, 247), and then we change the RS codes adaptively according to the channel condition of a time-varying wireless channel.

¹The active area efficiency is defined as (the number of active gates in a system) / (the number of total gates in a system).

Table 1. The number of operations in each stage of RS decoding.

	Addition	Multiplication	Inverse
Syndroms	$2t \cdot n$	$\bullet \{n \cdot t \cdot (2t - 1)\}$	-
Peterson Algorithm	$\frac{t}{2} \cdot (t - 1) \cdot (2t - 1) + t \cdot (t - 1)$	$\frac{t}{2} \cdot (t - 1) \cdot (2t - 1) + t \cdot (t - 1) + \frac{t}{2} \cdot (t + 1)$	$\frac{t}{2} \cdot (t + 1)$
Berlekamp Massey Algorithm	$(2t - 1) + 4t \cdot (2t - 1)$	$2t + 4t \cdot (4t - 1)$	-
Chien Search	$2t \cdot n$	$\ast \{n + n \cdot t \cdot (2t + 1)\}$	-
Error magnitudes	$\frac{t}{2} \cdot (t - 1) \cdot (2t - 1) + t \cdot (t - 1)$	$\frac{t}{2} \cdot (t - 1) \cdot (2t - 1) + t \cdot (t - 1) + \frac{t}{2} \cdot (t + 1)$	$\frac{t}{2} \cdot (t + 1)$
Error Correction	t	$\ast \{n\}$	-

*{Constant times of $\times \alpha, \times \alpha^{-1}$.}

For the adaptation of error correction capability t , we can utilize packet error rate(PER) as a threshold value to change t . Based on the PER, we can estimate the most effective error correction capability t which maximizes the throughput.

For reliable communication systems, we consider a reliable transport protocol which combine the adaptive FEC scheme and ARQ (Automatic Retransmission reQuest). In the protocol, if a receiver cannot correct the packets with FEC code, the receiver requests a retransmission. While the receiver waits for the retransmission packet, following arrival packets are placed in a received buffer. When the received buffer becomes full, eventually the following arrival packets will be dropped out. If packet processing throughput is improved at receivers for the reliable transport protocol, the receivers can process more packets per unit time, and can reduce the packet dropping rate from the received buffer.

We propose a reconfigurable adaptive FEC system. Our reconfigurable adaptive FEC system reconfigures dynamically an optimal RS decoder whose decoding throughput is maximized for error correction capability t in a limited hardware resource.

3. OPTIMIZATION OF RS DECODER

In this section we propose an optimization of RS decoder for each correction capability $t = 1, \dots, 4$ to realize reconfigurable adaptive FEC system.

3.1. RS Decoder Design for Each Error Correction Capability

An RS code is a block code which is defined over the Galois field $GF(2^m)$. We assume $m = 8$, and

(n, k) RS code has a length of $n = 2^8 - 1 = 255$ symbols which includes k symbols of information.

RS decoding is performed in 5 stages. First, the syndrome has to be calculated from the received codeword. Second, the Peterson or Berlekamp Massey[6],[7] algorithm has to be processed to get the error location polynomial. Third, the error locations are found by performing the Chien Search. Fourth, the error magnitudes are processed by the following equation. Fifth, by using the error values and the error locations, the codeword can be corrected.

Table 1 shows the number of additions, multiplications, and inverses in each stage. Note that n denotes the length of RS code, and t denotes an error correction capability. In the table, The number of operations in stages 2 and 4 is dependent on only error correction capability t . Therefore, we can optimize the function units for stages 2 and 4 by considering the number of operations for t , and we design the RS decoder which has 5-stage pipeline.

In stage 2 which computes error location polynomial, Peterson algorithm and Berlekamp Massey (BM) algorithm are typically employed. Decoders

which are based on the BM algorithm are proposed in [7],[6]. These decoders have a key-equation-solving block. The key-equation-solving block is a systolic architecture which has $3t$ processing elements, and can compute the error location polynomial in $2t$ clock cycles. However, since all the processing elements are composed of adders, multipliers and multiplexers, the hardware architecture has large number of gates. On the other hand, in the Peterson algorithm, the coefficients of an error location polynomial are derived from solving the matrix of syndroms, it follows that,

$$\begin{bmatrix} s_{t-1} & s_{t-2} & \dots & s_0 \\ s_t & s_{t-1} & \dots & s_1 \\ \vdots & \vdots & \ddots & \vdots \\ s_{2t-2} & s_{2t-3} & \dots & s_{t-1} \end{bmatrix} \cdot \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{t-1} \end{bmatrix} = \begin{bmatrix} s_t \\ s_{t+1} \\ \vdots \\ s_{2t-1} \end{bmatrix}. \quad (1)$$

Moreover, in stage 4 which computes error magnitudes, we also compute the matrix which is composed of error locations and syndroms, it follows that,

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ \alpha^{j_1} & \alpha^{j_2} & \dots & \alpha^{j_\nu} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{(t-1)j_1} & \alpha^{(t-1)j_2} & \dots & \alpha^{(t-1)j_\nu} \end{bmatrix} \cdot \begin{bmatrix} e_{j_1} \\ e_{j_2} \\ \vdots \\ e_{j_\nu} \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{t-1} \end{bmatrix}. \quad (2)$$

We employ the Peterson algorithm, and propose a matrix solver in order to solve these matrices by considering as follows.

1. Since the Eq.(1) and Eq.(2) are same form of matrix, we can share the matrix solver between stages 2 and 4. Therefore, we can reduce the hardware scale between these most complicated stages of the RS decoder.
2. In the Peterson algorithm, the matrix is solved by Galois field arithmetic operations step by step. Therefore, the matrix solver can be composed of less arithmetic function units than those for the systolic architecture with BM algorithm.
3. Although the clock cycles required to solve the matrix are more than those required by the systolic architecture with BM algorithm, we can apply the matrix solver to stages 2 and 4 within the limit of clock cycles. Since it takes 255 clock cycles to compute stages 1,3, and 5 respectively, we compose the matrix solver which satisfies the total clock cycles of stages 2 and 4 to be less than 255 for each error correction capability t .

We optimize the matrix solver for error correction capability $t = 1, \dots, 4$, and note that we employ LU Decomposition as the method of solving the matrix.

For error correction capability $t = 1$, Eq.(1) results in the equation of $s_0 \cdot \lambda_0 = s_1$. Therefore, the operation required in the stage 2 is only one division. Moreover, the Eq.(2) results in the equation

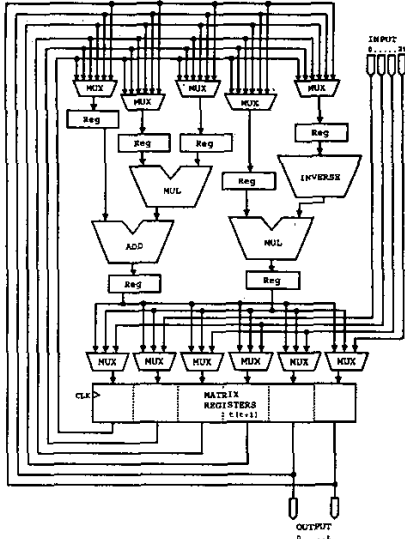


Fig. 1. A matrix solver($t = 4$).

of $e_j = s_0$. Therefore, no operation for stage 4 is required. We can consider that just one division for stage 2 is allowed to be computed within 255 clock cycles as well as the other stages. Thus, we can derive an element of inverse from evaluating $\alpha^i \cdot \alpha^j = 1$ for all elements of $\alpha^j (j = 0, 1, 2, \dots, n-1) \in GF(2^8)$. This can be performed by a multiplier and a LFSR, and this method can be realized by much smaller number of gates than the architecture for inverse which can be derived in one clock cycle.

For error correction capability $t = 2, 3$, the two $t \times t$ matrices of Eq.(1),(2) have to be calculated in stages 2 and 4 respectively. An arithmetic logic unit(ALU) enables to calculate these matrices in 255 clock cycles. The ALU is composed of an adder, a multiplier and an inverse over the $GF(2^8)$.

For error correction capability $t = 4$, we propose an optimal matrix solver shown in Fig.1. In the matrix solver a multiply-accumulate and a division is available, and the matrix register is composed of $t \cdot (t+1)$ words of 8bit for coefficients and variables. LU decomposition can write the lower triangular matrix L and higher triangular matrix U over the matrix register on its procedure.

3.2. Implementation of RS Decoder on FPGA

We implement RS decoders for error correction capability $t = 1, \dots, 4$. The proposed RS decoder is modeled in VHDL and implemented on Virtex II series FPGA provided from Xilinx. The device is XC2V1000 fg456-4 which has 5120 SLICES and the time of re-configuration is 9.36(ms)[9], then synthesis and implementation are carried out using ISE Ver.5.1i provided from Xilinx.

The delay of all the RS decoder is less than 10(ns), so that these RS decoder can achieve maximum frequency of more than 100(MHz).

The implementation SLICES result in 285 for $t = 1$, 651 for $t = 2$, 971 for $t = 3$, and 1389 for $t = 4$ per one RS decoder. Thus, we can implement 14-parallel RS decoder for $t = 1$, 6-parallel RS decoder for $t = 2$, 4-parallel RS decoder for $t = 3$, and 3-parallel RS decoder for $t = 4$ respectively on the FPGA.

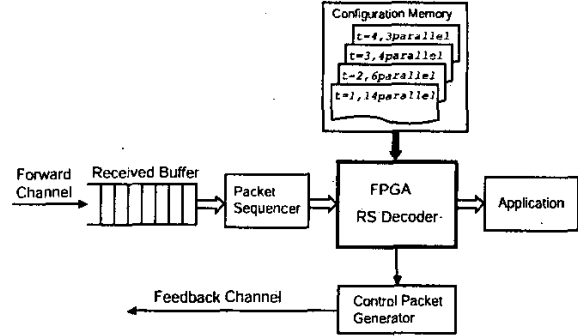


Fig. 2. A reconfigurable adaptive FEC system.

4. RECONFIGURABLE ADAPTIVE FEC SYSTEM

In this section, we propose a reconfigurable adaptive FEC system, then we evaluate packet dropping rate and data transmission throughput.

4.1. System Architecture

Fig. 2 shows our reconfigurable adaptive FEC system. Our system in Fig.2 is composed of the FPGA noted in Section 3.1, a configuration memory, a packet sequencer, a control packet generator, a received buffer, and an application.

Our reconfigurable adaptive FEC system reconfigures the RS decoder whose parallelism is maximized within the FPGA SLICES for each error correction capability t . This reconfiguration occurs dynamically whenever the error correction capability t is changed according to time-varying wireless channel condition. Configuration data of the RS decoder designed by our method for $t = 1, \dots, 4$ are stored in the configuration memory.

In our system, we assume that the application requires reliable packets which include no error. Arrival packets through wireless channel are placed in the received buffer. In reliable transport protocol, the sequence number of arrival packets has to be controlled by the packet sequencer.

The RS decoder operates the packets delivered from the packet sequencer. If the packets include no uncorrectable error, the RS decoder can output the correct packets to an application, then the RS decoder also outputs a receiving signal to the control packet generator. In this case the control packet generator creates positive acknowledgment (ACK) packet, then the receiver sends the ACK packet to the transmitter.

If the RS decoder cannot correct the packets which include uncorrectable errors, the RS decoder outputs an error signal to the control packet generator. In this case, the control packet generator creates a negative acknowledgment (NAK) packet which requests a retransmission, then the receiver sends the NAK packet to the transmitter. While receiver is waiting for the retransmission packet, following arrival packets are placed in the received buffer. Then, the following arrival packets remain in the received buffer until the error packets are recovered. When the received buffer becomes full, eventually the following arrival packets will be dropped out.

The control packet generator also creates a packet which requests a transmitter to change an error cor-

Table 2. Comparison of packet dropping rate.

RTT[ms]	10.0	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0	100.0
F-AFEC	0.000102	0.002926	0.002784	0.003987	0.004228	0.003793	0.007362	0.011152	0.012447	0.015108
R-AFEC	0.000044	0.002730	0.002581	0.003701	0.003962	0.003591	0.007112	0.010944	0.012223	0.014920

Table 3. Comparison of data transmission throughput.

RTT[ms]	10.0	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0	100.0
F-AFEC	0.985769	0.980059	0.980788	0.978635	0.978890	0.974090	0.963326	0.956693	0.956508	0.953389
R-AFEC	0.985793	0.980190	0.981056	0.979248	0.979340	0.974248	0.963672	0.956855	0.956802	0.953808

rection capability t . When the packets reflecting the new error correction capability t are delivered to the RS decoder, the receiver begins to reconfigure the RS decoder corresponding to t . Clearly, during the re-configuration, RS decoding process is suspended.

4.2. Experimental Results

The wireless channel can be approximately modeled by Gilbert model[1],[8]. We assume that the practical throughput of data transmission is 24(Mbps), and packet arrivals are modeled by Poisson distribution. Using the above model, we generate wireless errors and simulate data transmissions. In simulation, the packet length is fixed to 255(bytes) which is same as length of one RS code, and the size of received buffer is 1.53(Mbytes) in which 6000 packets can be placed.

We apply our reconfigurable adaptive FEC system to above model. Assuming that clock frequency of the RS decoder is 10(MHz) for practical use, the decoding throughput per one RS decoder results in 78(Mbps). Therefore, the each parallel RS decoder which is optimized by our method achieves 1092(Mbps) for $t = 1$, 468(Mbps) for $t = 2$, 312(Mbps) for $t = 3$, 234(Mbps) for $t = 4$ respectively, and each decoding throughput can be achieved on the same FPGA.

For comparison, we apply our adaptive FEC scheme to a fixed hardware system. The fixed hardware system means that the hardware architecture of the RS decoder is not optimized for each error correction capability $t = 1, \dots, 4$. This system can operate as $t = 1, \dots, 4$, but its parallelism is to be fixed. Here, we assume that the RS decoder for $t = 4$ designed by our method is implemented fixedly on the FPGA, so that the parallelism of this RS decoder is fixed to 3, and it only achieves decoding throughput of 234(Mbps) regardless of t .

We evaluate the packet dropping rate. Hereinafter, *R-AFEC* denotes our reconfigurable adaptive FEC system, and *F-AFEC* denotes the adaptive FEC scheme on the fixed hardware system. Evaluation results are shown in Table 2. In the table, Round Trip Time (RTT) is the time between transmitting a packet and receiving its response. We assume that the RTT is 10(ms), \dots , 100(ms). The table shows that the packet dropping rate of both systems increases depending on the RTT. This is because it takes more times to recover error packets by ARQ according to increase of the RTT. The packet dropping rate on the R-AFEC is decreased by a maximum of 43% compared to the F-AFEC.

We also evaluate the data transmission throughput of the R-AFEC and F-AFEC. The evaluation results of the data transmission throughput are shown in Table 3. The table shows that our reconfigurable adaptive FEC achieves better throughput than conventional fixed adaptive FEC. This is because our reconfigurable adaptive FEC can reduce the packet

dropping rate more efficiently than the fixed hardware system in addition to error correction capability t can be changed adaptively according to the wireless channel condition.

5. CONCLUSION

In this paper, we proposed a reconfigurable adaptive FEC system. Our reconfigurable adaptive FEC system reconfigures dynamically an optimal RS decoder whose decoding throughput is maximized for error correction capability t within a limited hardware resource. Our reconfigurable adaptive FEC system can reduce packet dropping rate by a maximum of 43% in comparison with conventional fixed hardware systems, and also we can improve data transmission throughput for a reliable transport protocol. We will propose a new reconfigurable processor for the fast wireless communication systems in the future.

Acknowledgment

This work was supported by fund from the MEXT via Kitakyushu innovative cluster project.

6. REFERENCES

- [1] A. Chockalingam, M. Zorzi, and R. R. Rao, "Performance of TCP Reno on Wireless fading links with memory," *Proc. IEEE ICC'98*, vol. 2, pp. 595-600, June 1998.
- [2] Daji Qiao, Shin, K.G., "A two-step adaptive error recovery scheme for video transmission over wireless networks," *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proc. IEEE INFOCOM 2000*, Vol.3, pp.1698-1704, March 2000.
- [3] Sungrae Cho, Goulart A., Akyildiz I.F., Jayant N., "An adaptive FEC with QoS provisioning for real-time traffic in LEO satellite networks," *IEEE International Conference on Communications, ICC 2001*, Vol.9, pp.2938 - 2942, June 2001.
- [4] D.V.Sarwate and N.R.Shanbhag, "High-speed architecture for Reed-Solomon decoders," *IEEE Trans on VLSI Systems*, vol.9(5), pp.641-655, Oct.2001
- [5] Fraser Cameron, Moshe Zukeman, Maxim Gitlits, "Adaptive Transmission Parameters Optimization in Wireless Multi-Access Communication," *IEEE International Conference on Networks*, pp.91-95, Oct.1999.
- [6] J-H. Jeng and T.K.Truong, "On decoding of both errors and erasures of a Reed-Solomon code using an inverse-free Berlekamp-Massey algorithm". *IEEE Trans. on Communications*, vol.47(10):pp.1488-1494, Oct. 1999.
- [7] Trieu-Kien Truong and K. C. Hung, "Inversionless decoding of both errors and erasures of Reed-Solomon code," *IEEE Trans. on Communications*, vol.46, pp.973-976, Aug. 1998.
- [8] Youshi Xu, Tingting Zhang, "An Adaptive Redundancy Technique for Wireless Indoor Multicasting," *Fifth IEEE Symposium on Computers and Communications(ISCC 2000)*, pp.607-615, Jul. 2000.
- [9] Xilinx, Inc., Vertex-II Platform FPGA User Guide, UG002(v1.6.1), p.293, Aug, 2003.