An Early Termination Criterion for Stochastic LDPC Decoding

Huiming Dai, Qiangze Chi, Hao Yan, Xin Zhang, and Dacheng Yang
Beijing University of Posts and Telecommunications, Beijing, China
Email: davidhmdai@gmail.com, zhangxin@bupt.edu.cn

***Abstract*—Stochastic decoding is an excellent approach for Low-Density Parity-Check (LDPC) codes which are adopted in many communication standards, including 10GBASE-T, DVB-S2, WiMAX. In this paper, we first research some novel schemes of stochastic LDPC decoding, like Noise-Dependent Scaling (NDS), Edge Memories (EMs), Tracking forecast memories (TFMs) and majority-based tracking Forecast Memories (MTFMs). (7, 4) Hamming code and (2048, 1723) LDPC code from the IEEE 802.3an standard are used to simulate and analyze the advantages and disadvantages of these methods. Then, an early termination criterion is proposed as an efficient way to speed up the decoding procedure. Simulation results show that our method can decrease the computation complexity significantly with negligible performance loss.**

***Index Terms*—Low-Density Parity-Check (LDPC), stochastic decoding, Edge Memories (EMs), low complexity, early termination criterion**

I. Introduction

In 1963, LDPC codes were first proposed by Gallager [1], which are an important class of linear block codes. However, because of limited technical conditions and lack of feasible decoding algorithm, LDPC codes were ignored at that time. Not until the proposal of Tanner graph [2] and the discovery of Turbo codes [3] did people restart the research of LDPC. After several years of study and development, breakthroughs were made in many aspects [4]-[6]. It has been demonstrated that LDPC codes can provide decoding performance close to the Shannon capacity limit [7]. For these reasons, LDPC codes have been adopted for several communication standards, including IEEE 802.3an (10GBASE-T) [8], IEEE 802.16 (WiMAX) [9], IEEE 802.11 (Wi-Fi) [10] and the digital video broadcasting (DVB-S2) standards [11].

Although LDPC codes have excellent coding performance, the high-complexity hardware architecture is the major challenge of LDPC decoder in actual applications. In recent years, stochastic decoding algorithms are proposed which can be used to implement the hardware of LDPC decoders [12]. And FPGA-based stochastic architectures that decode LDPC codes are realized in [6], [13]. It has also been applied for decoding non-binary LDPC codes in [14]. Stochastic decoding uses Bernoulli bit stream to represent the probability messages. The operations in the probability domain can be converted into the brief bitwise operations by stochastic computation. Therefore stochastic decoding offers an affordable solution for fully parallel implementation of medium and long codes.

However, stochastic decoding may encounter the latching problem, especially in long LDPC codes [15]. The latching problem is that Variable Nodes (VN) cannot be updated effectively in several decoding iterations because of the cycles in a factor graph. It results in the slow convergence of decoding procedure and degrades the decoding performance.

To combat this problem, Edge Memories (EMs) and Noise-Dependent Scaling (NDS) are introduced [12]. NDS uses a scaling method to increase the level of switching activity over different ranges of SNRs. EMs utilizes M-bit shift register to re-randomize the stochastic bit streams, so that it can reduce the chance of latching obviously. And Tracking Forecast Memories (TFMs) is proposed [16] which can provide similar decoding performance to EMs while having lower hardware complexity.

As we know, EMs and TFMs are assigned to each outgoing edge of a VN and consume considerable silicon area in ASIC stochastic decoders. In order to reduce more hardware complexity than EMs and TFMs, a new stochastic approach based on Majority-based Tracking Forecast Memories (MTFMs) is proposed in [17] for decoding LDPC codes. A VN needs only one MTFM so that the implementation of LDPC decoders can be much simpler than previous approaches.

In this paper, we propose a new Early Termination Criterion for LDPC stochastic decoding. Normally, the decoding cycles are set very high for the sake of precision. However, as we have researched, most checksums will converge in advance, thus making it unnecessary to proceed the remaining cycles. In our scheme, convergence points are obtained adaptively to different bit streams so that we could know when to stop the decoding procedures. With this new termination criterion, decoding cycles can be reduced significantly. Simulation results demonstrate that our proposed scheme outperforms conventional ones in computation complexity while losing little performance. The rest of this paper is organized as follows. In Section II we review the fundamentals of stochastic decoding. Section III describes approaches (NDS and EMs) to solve latching

Manuscript received December 21, 2015; revised June 16, 2016.

doi: 10.18178/lnit.4.1.7-12

problem and the low complexity approaches (TFMs and MTFMs). Section IV proposes the Early Termination Criterion, discusses different situations of the convergence and applies it to the implementation of LDPC decoding. Then, simulation results of the above methods are presented in Section V. Finally, Section V gives the conclusions.

II. The Principle of Stochastic Decoding

In Sum-Product Algorithm (SPA) belief messages passed between variable nodes and check nodes are in the form of probabilities with values between 0.0 and 1.0. Because of the complex calculation in the probability domain, it will require much hardware resources.

Stochastic computation is a novel approach for decoding LDPC codes. In this approach, probabilities received from the channel are converted to Bernoulli sequences. The frequency of '1's in a stream of bits represents a probability of belief message. For example, a probability of 0.2 can be converted to a stream of bits whose 20% of bits are '1's. The transformation of a probability to a stochastic sequence is non-unique, which means that the order of `1's in a stochastic stream is not important.

Stochastic decoding is an iterative-based decoding algorithm. In one iteration, belief messages are passed from the variable nodes to the check nodes along the interconnected edges, and then returned from the check nodes to the variable nodes. The operation of belief propagation between two types of nodes can be done by simple logic gates [4].

Let $P_a = \Pr(a = 1)$ and $P_b = \Pr(b = 1)$ be the probabilities of two inputs in a check node whose degree is 3. The output probability of the check node is:

$$P_c = P_a(1-P_b) + P_b(1-P_a) \tag{1}$$

In a degree-2 variable node, the output message is:

$$P_c = \frac{P_a P_b}{P_a P_b + (1-P_a)(1-P_b)} \tag{2}$$

When implemented on the hardware, a check node needs only an XOR gate. The multiplication of two values can be performed by an AND gate and the division of two values can be approximated by a JK flip-flop. Fig. 1 shows the hardware structures for (1) and (2).

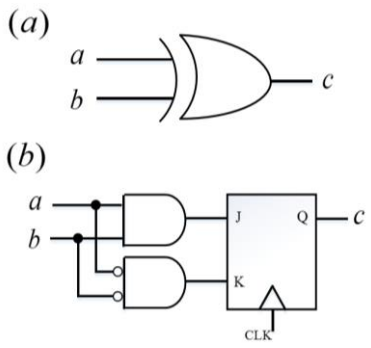

Figure 1. The structures of (a) PN and (b) VN in stochastic decoders.

The operation of a variable node is something more complex than a check node. When all inputs of a variable node are not equal, we must figure out what to output at the decoding cycle. For example, when a variable node holds on the previous bit on the edge, this is referred to a hold state of the variable node. When all the inputs are equal, we call it non-hold state.

However, there is a challenge for practical application with these structures [15]. When a cycle in the factor graph exists, some variable nodes are prone to lock into a fixed state. Once the variable nodes cannot be updated effectively in several decoding iterations, it will lead to slow convergence and low performance of decoding.

III. Novel Algorithms for Stochastic Decoding

A. Combating the Latching Problem

As described in Section II, a stochastic decoder is very sensitive to the level of random switching activity, there is possibility that it runs into a latching problem [18]. In [12], two approaches are proposed to deal with the latching problem. One approach is Noise-Dependent Scaling (NDS), the other is Edge Memories (EMs). NDS increases the level of switching activity in stochastic streams over different SNRs, the received channel LLRs are scaled by a factor which is proportional to the noise level in the channel.

Assuming a BPSK transmission over an AWGN channel, the scaled LLR L_i' for the i-th symbol y_i in the received block is calculated as:

$$L_i' = (\frac{\alpha N_0}{Y})L_i = \frac{4\alpha y_i}{Y} \tag{3}$$

where N_0 is the noise power spectral density, $L_i = 4y_i / N_0$ is the channel LLR for y_i, Y is a fixed maximum value of the received symbols. For example, for a BPSK modulation Y can be set at 6. And α is also a constant factor whose value is chosen based on the best BER performance.

When the hold state occurs, it is not a good idea to output the bit from the last non-hold state all the time. The use of EMs is a good way to deal with this problem. An EM is assigned to every edge of VN which can provide the approximate information when the hold state occurs.

Bits from VN can be classified into two types: regenerative bits and conservative bits [13]. Only regenerative bits can correctly reflect the belief messages from CN because it is generated when a VN is in the non-hold state. When the non-hold state occurs, a VN outputs the newly generated regenerative bit and EM is updated with the regenerative bit. When the hold state occurs, the VN outputs a bit randomly chosen from the relevant EM.

An EM can be implemented by an M-bit shift-register, which stores no more than M regenerative bits. So it is for sure that VN generates a proper bit relying on the most recent regenerative bits when it is in the hold state.

NDS and EMs can work together in stochastic decoding for LDPC codes [19]. Section V will show the simulations of decoding with NDS and EMs.

B. Reduced Hardware Complexity

Though EMs have good decoding performance, they occupy much silicon area when implemented in ASICs. In order to reduce the hardware complexity, Tracking Forecast Memories (TFMs) and Majority-based tracking forecast memories (MTFMs) is proposed in [16], [17].

TFMs are similar to EMs for using information from the previous reliable regenerative bits to generate approximate bits when VN is in the hold state. A TFM hold a probability of stochastic streams from VN and recursively updates it based on the previous regenerative bits.

Let $b(t)$ be the regenerative bit from a VN and $\mathrm{P}(t)$ be the probability kept by a TFM. When the non-hold state occurs, $\mathrm{P}(t)$ is updated as follows:

$$P(t+1) = (1-\beta(t))P(t) + \beta(t)b(t) \tag{4}$$

where $\beta(t)$ is the coefficient and $0 < \beta(t) < 1$. When the hold state happens, $\mathrm{P}(t)$ is compared to a (pseudo) random number $\mathrm{R}(t)$, and the new bit $b'(t)$ is generated as follows:

$$b'(t) = \begin{cases} 1 & \text{for } P(t) > R(t) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

TFMs are assigned to every edge of VN. The workflow of one edge with TFM is shown in Fig. 2.

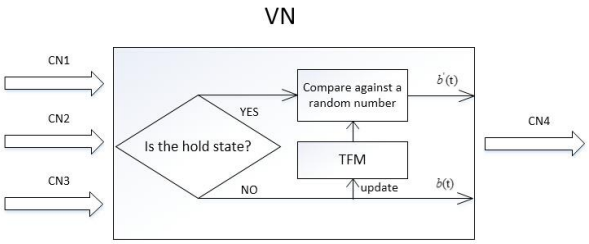

Figure 2. The workflow of one edge with TFM.

Different from EMs and TFMs that a VN uses one memory in every edge, only one MTFM is needed in each VN. The algorithm of MTFMs has a different update rule from that of TFMs, it is based on majority of regenerative bits of VN. As we know, there are several edges in a VN. For some edges, it may be in the hold state; for others, it is in the non-hold state. The MTFM is updated only if a certain percentage of edges are in the non-hold state at least. Let $S(t)$ be the number of edges which are in the non-hold state and T_u is referred as a threshold to update an MTFM.

$$S(t) > T_u \tag{6}$$

When more than half of the regenerative bits are '1's, we believe that VN's regenerative bits $b(t)$ are '1's for any edge which is in the hold state.

The operations of VN with MTFMs are the following:

1) When an edge of VN is in the non-hold state, it directly uses the corresponding regenerative bit as the outgoing bit of the edge.
2) When an edge is in the hold state, the MTFM generates "regenerative bits" for the edge. The edge outputs the "regenerative bits" as the outgoing bit.

MTFMs significantly reduce the hardware complexity than TFMs. However, as MTFMs generate the same regenerative bit for all edges which are in the hold state, while, actually, different edges have different probabilities to be '1', so the decoding performance of MTFMs is not as good as TFMs. Section V will show the simulation of decoding with TFMs and MTFMs.

IV. Early Termination Criterion

Normally, there are two termination criteria for stochastic decoding: (1) if all the PNs are satisfied or (2) if a maximum number of DCs has been exceeded. Decoder outputs the sign-bit of each up/down counter as the decoded codeword when either criteria is satisfied.

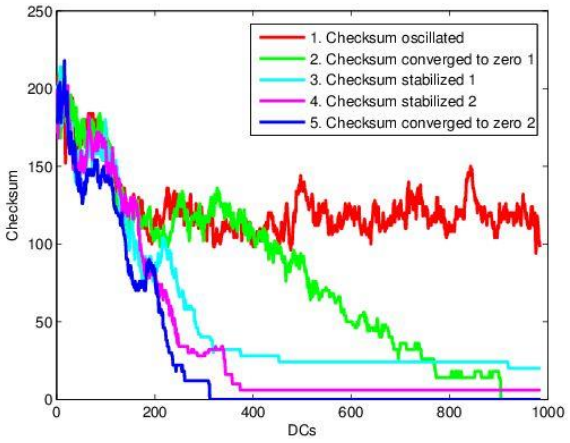

Figure 3. Checksums of different decoding bits in a LDPC frame, the order of the curves is top-down.

In Sum-Product Algorithm, some research has been done on the change regulations of LLRs' values in VNs and PNs, and stopping schemes are made to decrease iteration numbers [20]-[22]. As stochastic computation has high accuracy, we also believe that there is no need for all decoding procedures to finish the maximum DCs. According to our statistics shown in Fig. 3 that demonstrates checksum of all the check nodes in a LDPC frame, we can see that some checksums (Curve 2, Curve 3, Curve 4, Curve 5) are in downward trends, and become stable (0 or some fixed value) at certain DCs. After that, the decoding result will not change. There are five curves in the figure, whose convergence speeds are not the same. So we can adaptively stop decoding in advance for different LDPC frames, which will reduce the amount of calculation greatly.

Our proposed early termination criterion is shown in Fig. 4. For decoded frames whose checksums can be stabilized to a value for at least "max" times, we should stop the procedure in advance; otherwise, we should continue decoding until the number of DCs reaches

maximum. Of course this may not always be right, it is a compromise between complexity and performance.

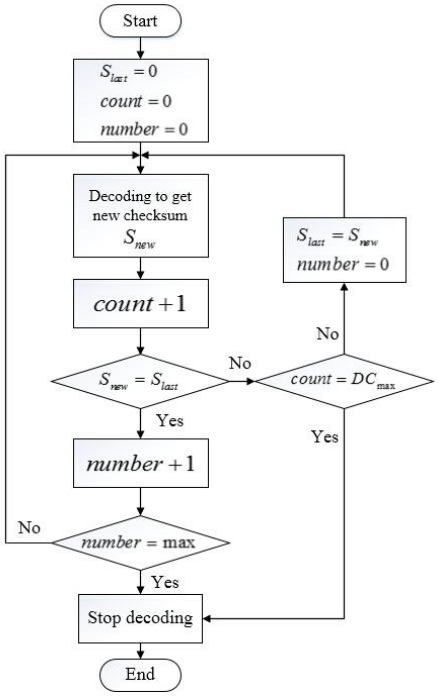

Figure 4. Early termination criterion for stochastic LDPC decoding.

V. SIMULATION

A. Simulation Results of NDS and EMs

As analyzed in Section III, the addition of NDS and EMs can significantly improve the effect of traditional stochastic decoding which is affected by the latching problem. For simplicity, we choose (7, 4) Hamming code to simulate the performance of BER. The maximum number of DCs is 2K. The iteration number in SPA is 10. DCs are not equivalent to the iterations in SPA. The length of EMs is 32.

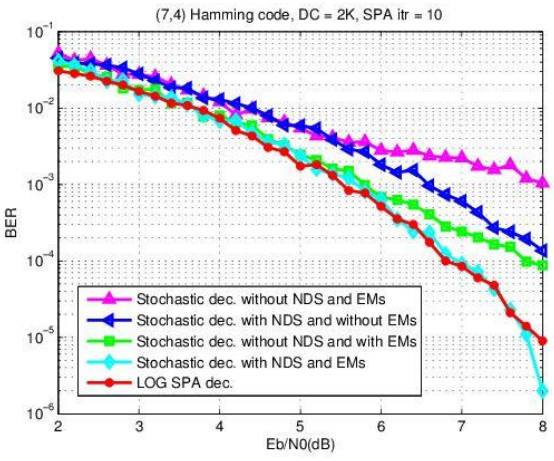

Figure 5. Simulation results for (7, 4) hamming code using NDS and EMs.

We can see in Fig. 5 that both of NDS and EMs used in stochastic decoding can achieve a better performance than traditional stochastic decoding. In order to see the impacts of NDS and EMs respectively, stochastic decoding with NDS and without EMs and stochastic decoding without NDS and with EMs are simulated for comparison. The simulation results show that EMs contribute more to the performance than NDS. We also simulate the stochastic decoding with both NDS and EMs, the result show that the combined scheme has comparable BER performance to floating point SPA decoding algorithm.

B. Simulation Results of TFMs, MTFMs

We choose (2048, 1723) LDPC code which is used in the IEEE 802.3an (10GBASE-T) standard to simulate the two means. The maximum number of DCs is 10K. The iterations of SPA is 10. The length of EMs is set to 32. The parameters of TFMs and MTFMs are the same as [16], [17].

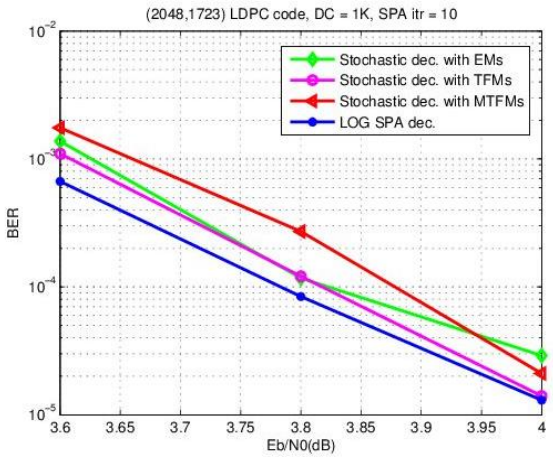

Figure 6. Simulation results for (2048, 1723) LDPC code using TFMs and MTFMs.

Fig. 6 shows BER performance for a (2048, 1723) LDPC code using low-complexity stochastic decoding. As shown, TFMs performs a little better than EMs, while the hardware complexity of it is simpler than EMs. Though at $BER=10^{-4}$, the performance loss of MTFMs compared to EMs is about 0.03dB, a VN needs just only one MTFM for all edges. So it is much easier for MTFMs to be implemented.

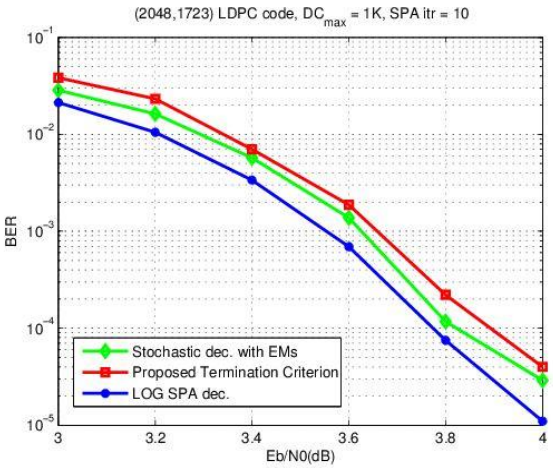

Figure 7. Simulation results for (2048, 1723) LDPC code using proposed termination criterion.

C. Simulation Results of Proposed Termination Criterion

Simulation parameters are the same as the above. From Fig. 7, we can see that our proposed termination criterion can adapt to different convergence situations well. At BER of 10^{-4}, our scheme loses only 0.07dB than EMs, while the average decoding speed is 63% faster. Considering the great amount of computation it reduces, the compromise is worthy.

VI. Conclusions

In this paper, we have presented a tutorial overview of some novel methods for decoding state-of-the-art LDPC codes, like NDS, EMs, TFMs and MTFMs. By realizing these methods integrally, we can know the advantages and disadvantages of them. Besides, it also helps calibrating our simulation ways and platform, which contributes to the correct implementation of our new method. And then an early termination criterion for stochastic LDPC decoding is proposed. Theoretical analysis and simulation results indicate that our proposed scheme is able to provide similar decoding performance as traditional ones while having much less computation. Some further comparisons and association schemes will be researched in the next step.

References

[1] R. G. Gallager, "Low-Density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21-28, 1962.

[2] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533-547, 1981.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-Codes. 1," in *Proc. IEEE International Conference on Communications*, May 1993, pp. 1064-1070.

[4] V. C. Gaudet and A. C. Rapley, "Iterative decoding using stochastic computation," *Electronics Letters*, vol. 39, no. 3, pp. 299-301, Feb. 2003.

[5] A. Rapley, C. Winstead, V. Gaudet, and C. Schlegel, "Stochastic iterative decoding on factor graphs," in *Proc. 3rd Int. Symp. Turbo Codes Related Topics*, 2003, pp. 507-510.

[6] W. J. Gross, V. C. Gaudet, and A. Milner, "Stochastic implementation of LDPC decoders," in *Proc. Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, Oct. 2005, pp. 713-717.

[7] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, no. 18, Aug. 1996.

[8] IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks—Specific Requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, IEEE Standard P802.3an/D4.0, 2006.

[9] IEEE Standard for Air Interface for Broadband Wireless Access Systems, IEEE Standard 802.16-2012, Aug. 2012.

[10] IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks—Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Standard 802.11-2012, Mar. 2012.

[11] Digital Video Broadcasting (DVB); Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications (DVB-S2), European Telecommunications Standards, Mar. 2013.

[12] S. S. Tehrani, W. J. Gross, and S. Mannor, "Stochastic decoding of LDPC codes," *Communications Letters, IEEE*, vol. 10, no. 10, pp. 716-718, 2006.

[13] S. S. Tehrani, S. Mannor, and W. J. Gross, "An area efficient FPGA-based architecture for fully-parallel stochastic LDPC decoding," in *Proc. IEEE Workshop on Signal Processing Systems*, 2007, pp. 255-260.

[14] G. Sarkis, S. Mannor, and W. J. Gross, "Stochastic decoding of LDPC codes over GF (q)," in *Proc. IEEE International Conference on communications*, 2009, pp. 1-5.

[15] C. Winstead, V. C. Gaudet, A. Rapley, and C. Schlegel, "Stochastic iterative decoders," in *Proc. IEEE International Symposium on Information Theory*, 2005, pp. 1116-1120.

[16] S. S. Tehrani, A. Naderi, G. A. Kamendje, S. Mannor, and W. J. Gross, "Tracking forecast memories for stochastic decoding," *Journal of Signal Processing Systems*, vol. 63, no. 1, pp. 117-127, 2011.

[17] S. S. Tehrani, A. Naderi, G. A. Kamendje, S. Hemati, S. Mannor, and W. J. Gross, "Majority-Based tracking forecast memories for stochastic LDPC decoding," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4883-4896, 2010.

[18] K. L. Huang, V. Gaudet, and M. Salehi, "A scaling method for stochastic LDPC decoding over the binary symmetric channel," in *Proc. 47th Annual Conference on Information Sciences and Systems*, 2013, pp. 1-5.

[19] S. S. Tehrani, S. Mannor, and W. J. Gross, "Fully parallel stochastic LDPC decoders," *IEEE Transactions on Signal Processing*, vol. 56, no. 11, pp. 5692-5703, 2008.

[20] F. Kienle and N. Wehn, "Low complexity stopping criterion for LDPC code decoders," in *Proc. IEEE 61st Vehicular Technology Conference*, 2005, pp. 606-609.

[21] J. Li and X. H. You, "Early stopping for LDPC decoding: Convergence of Mean Magnitude (CMM)," *IEEE Communications Letters*, vol. 10, no. 9, pp. 667-669, 2006.

[22] Z. Cai, J. Hao, and L. Wang, "An efficient early stopping scheme for LDPC decoding based on checknode messages," in *Proc. 11th IEEE Singapore International Conference on Communication System*, Nov. 2008, pp. 1325-1329.

Huiming Dai is a Ph.D. student in Wireless Theories and Technologies Lab (WT&T), Beijing University of Posts and Telecommunications (BUPT), China. He received his B.E. degree of Communication Engineering in BUPT in 2012. His research now is focused on wireless communications, especially on vehicular communications. The author's email is davidhmdai@gmail.com.

Qiangze Chi is a graduate student in Wireless Theories and Technologies Lab (WT&T), Beijing University of Posts and Telecommunications (BUPT), China. He received his B.E. degree of Communication Engineering in BUPT in 2014. His research now is focused on wireless communications, especially on interference cancellation.

Hao Yan is a graduate student in Wireless Theories and Technologies Lab (WT&T), Beijing University of Posts and Telecommunications (BUPT), China. He received his B.E. degree of Communication Engineering in BUPT in 2013. His research now is focused on wireless communications, especially on channel equalization.

Xin Zhang received Ph.D. degree in communication engineering from BUPT in 2002, and is currently an associate professor in WT&T Lab at BUPT. His major research interest is focused on theories and technologies of wireless communications. As a leader or a key member, he has participated in many research programs, and has published more than 30 papers in international journals, as well as other 5 books.

Dacheng Yang received Ph.D. degree in communication engineering from BUPT in 1988, and is currently the professor and director in WT&T Lab at BUPT. Since 1988, he has engaged in studies on communication systems and he is now the Chinese chief expert of BUPT-QUALCOMN Joint Research Center. His recent interests are in wireless transmission techniques and systems.