Enhanced Verification-Based Decoding for Packet-Based LDPC Codes over Wireless Channels

Bin Zhu∗†, Defeng (David) Huang† and Sven Nordholm∗
∗Western Australian Telecommunications Research Institute (WATRI)
39 Fairway, Crawley WA, Australia 6009
email: {zhub, sven}@watri.org.au
†School of Electrical, Electronic and Computer Engineering
University of Western Australia, Crawley WA, Australia 6009
email: huangdf@ee.uwa.edu.au

Abstract—In this paper, we apply the enhanced verification-based decoding algorithm (EVA) for packet-based LDPC codes over wireless channels. Compared with the verification algorithm (VA) in the literature, the EVA algorithm enhances the verifica-
tion condition thereby reducing the likelihood of false verification. The wireless channels in simulations are modeled by binary symmetric channel (BSC) and Gilbert-Elliott (GE) channel. The numerical results indicate the proposed algorithm gives a superior performance but with only moderate computation increase compared with VA. For example, when the bit error probability is less than 10−2 in bad state of GE channel, the EVA reduces the frame error rate (FER) by one order with less than 35% complexity increase compared with VA.

I. INTRODUCTION

Recent developments in wireless communications have resulted in an increasing demand for high data rate multimedia services. In this regard, wireless systems are required to support high speed packets transmission efficiently and reliably over hostile wireless channels. Channel coding is essential to improve reliability of communications. To this end, low-density parity-check (LDPC) codes [1] have become a powerful tool due to their superior channel-capacity achieving performance.

Research on LDPC codes [1]–[4] has traditionally focused on small alphabets. With the development of communication networks and high speed wireless systems, the operation units are normally blocks of bits organized as packets. As a result, some efforts have been made both in code design and decoding algorithms of packet-based channel codes [5]. For packet erasure channel, the Fountain codes such as LT-codes [6] and Raptor codes [7] have been proposed to be used for realizing multi-cast and content delivering networks. As for packet based LDPC codes, verification based decoding approach (VA) has been proposed by Luby and Mitzenmacher [8]. Their suggested decoding algorithm consists of two iterative stages: verification and correction. In the verification stage, if the sum of all the neighboring variable nodes of a check node equals zero, all these variable nodes are verified. In the correction stage, one unverified variable node can be corrected if all its other neighboring variable nodes incident to a check node have been verified. Its updated value is computed as the sum of the other neighbor variable nodes. However one limitation with VA is the way that the verification is done. There is a high likelihood of false verification because whether a variable node is verified or not depends only on one check node, which is not sufficient as will be shown in this paper by simulations.

In this paper, we propose to use the enhanced verification-based decoding algorithm (EVA) [9] for packet-based LDPC codes over wireless channels, which reduces the probability of false verification with only moderate computation complexity increase. Unlike the VA, EVA verifies each variable node based on multiple incident check nodes, the number of which is decreased from a maximum value to one along with the decoding process. Compared with the VA algorithm, the verification of EVA is more reliable for variable nodes. As a result, the approach gives a better decoding performance. Further, the VA can be considered as a specific case of EVA when the maximum number of check nodes required to be satisfied for a variable node is set to be one. Since EVA is a packet-based decoding algorithm, the packet-level operations are dominant in decoding complexity. The proposed EVA maintains the total number of packet-level operations in the same order as VA. To achieve better trade-off between performance and complexity, a simplified enhanced verification algorithm has also been proposed.

We use simulations to show the superior performance of EVA over wireless channels. For all packet sizes and channels tested, we find that EVA performs better than VA. For binary symmetric channel (BSC), EVA can achieve at least 100 times lower in frame error rate (FER) when crossover probabilities $p \leq 10^{-3}$, while its computation increases less than twice compared with VA. We also use Gilbert-Elliott (GE) channel [10], [11], a discrete time two-state Markov model, to approximate the wireless fading channel. Over GE channel, EVA exhibits at least 10 times better performance in FER than VA with a limited increase of computation loads.

This paper is organized as follows. In Section II, we describe the system model of packet-based LDPC codes. In Section III, we present the EVA for both regular and irregular LDPC codes.
codes and evaluate the computation complexity. In Section IV, we present simulation results for VA, EVA and simplified EVA for both BSC and GE channels. Finally, this paper is concluded in Section V.

II. SYSTEM MODEL

Let $x := (x_1, x_2, ..., x_n)$ be an input codeword which is encoded with an LDPC parity check matrix $H$ and $n$ is the length of the codeword, where $x_i := [x_i^1, x_i^2, ..., x_i^b]$ is a packet of a string of $b$ bits. The codeword $x$ is transmitted over a channel. At the output of the channel, $y := (y_1, y_2, ..., y_n)$ is received, where $y_i := [y_i^1, y_i^2, ..., y_i^b]$.

The LDPC code can be represented by a bipartite graph. The variable nodes represent the packets of the codeword and the check nodes determine the constraint that the sum of adjacent neighboring variable nodes equals zero, based on the parity check matrix $H$. Here the sum operation is a bitwise EXCLUSIVE-OR between packets.

LDPC codes can be classified into regular LDPC codes and irregular LDPC codes. In a regular LDPC code, the degree of variable nodes and the degree of check nodes are constant, which are denoted as $d_v$ and $d_c$, respectively. For irregular LDPC codes, the degrees of both variable nodes and check nodes are no longer constant but governed by a degree distribution. We use degree distribution polynomials [12] $\lambda(x)$ and $\rho(x)$ to specify degree distributions of variable nodes and check nodes, respectively. The degree distribution polynomial of variable nodes $\lambda(x)$ is defined as follows:

$$\lambda(x) = \sum_{d=1}^{d_{\text{max}}} \lambda_d x^{d-1}$$

where $\lambda_d$ denotes the fraction of all edges connected to degree-$d$ variable nodes and $d_{\text{max}}$ denotes the maximum degree of variable nodes. Similarly, the degree distribution polynomial of check node $\rho(x)$ is defined as follows:

$$\rho(x) = \sum_{d=1}^{d_{c,\text{max}}} \rho_d x^{d-1}$$

where $\rho_d$ denotes the fraction of all edges connected to degree-$d$ check nodes and $d_{c,\text{max}}$ denotes the maximum degree of check nodes.

III. ENHANCED VERIFICATION BASED DECODING ALGORITHM

A. Verification and Correction

The concepts of verification and correction were firstly proposed for packet-based LDPC codes by Luby et al. [8]. With each variable node, there are three possible values: a received value $r_v$, a current value $c_v$ and a final value $f_v$.

For verification, there are two important concepts in this paper: a variable node is verified by a specific check node and a variable node is verified.

1) A variable node $v_k$ is verified by a check node $c$ if the following is satisfied.

$$\sum_{v_k \in V_c} c_{vk} = 0 \quad (1)$$

where $V_c$ is the set of neighboring variable nodes linked with check node $c$.

2) A variable node is deemed verified if it is verified by a preset number of incident check nodes.

We note here in [8] the concept of a variable node is verified and that of a variable node is verified by a specific check node are indistinguishable since in VA, the variable node is verified as long as it is verified by one check node.

For correction, an unverified variable node can be corrected if all its other neighboring variable nodes incident to a check node have been verified. Specifically, let $\Omega$ denote the set of verified neighboring variable nodes of a check node $c$ and $v_l$ is the only unverified variable node incident to check node $c$.

The correction of variable node $v_l$ is achieved as follows:

$$f_{vl} = c_{vl} = \sum_{v_j \in \Omega, v_j \neq v_l} c_{vj} \quad (2)$$

B. EVA Algorithm

In VA algorithm, if a check node satisfies the constraint (i.e., the sum of its incident variable nodes equals zero), all its incident variable nodes are verified. As a result, there is a high likelihood that a variable node is incorrectly verified, thereby inducing a high frame error rate. In order to reduce the likelihood of such false verification of variable nodes, we enhance the verification requirement by requiring multiple constraints to be satisfied before a variable node can be claimed to be verified. In EVA, there are two possible states for variable nodes: either verified or unverified. Let $C_{\text{ver}}$ and $C_{\text{unver}}$ be the sets storing all verified and unverified variable nodes, respectively.

As shown in Algorithm 1, the EVA starts with all variable nodes being unverified. Correspondingly, all current values of variable nodes equal their $r_v$s and these variable nodes are put into $C_{\text{unver}}$. In the following part, the EVA iteratively decodes the received codewords under different thresholds. Let the set of threshold values denoted as $S = \{\gamma_1, \gamma_2, ..., \gamma_M\}$, where $\gamma_1 > \gamma_2 > ... > \gamma_M$. A confidence ratio is introduced to adjust the condition that a variable node is verified during the progress of EVA. The confidence ratio $\zeta$ is defined as $\frac{c_{vl}}{d_{vl}}$, where $\gamma_l \in S$ and $\gamma_{\text{max}}$ is a constant value. In the meantime, if a variable node $v_l$ is verified by $\text{num}_c$ check nodes, its relative ratio is $\frac{\text{num}_c}{d_{vl}}$, where $d_{vl}$ is the degree of the variable node $v_l$.
There are two iterative stages named verification and correction. For the \(i\)th round, in the verification stage, under certain threshold \(\gamma\), a variable node is verified if its relative ratio is not less than the confidence ratio. For example, for regular LDPC codes, if setting \(\gamma_{\text{max}} = d_v\), the equivalent condition is that a variable node is verified if it is verified by \(\gamma\) or more check nodes. Hereafter, its current value is fixed as its final value, otherwise it remains unchanged both in state and value. As for the second stage correction, for each check node, if all its other connected variable nodes have been verified except one, this variable node is corrected and finalized by substituting the current value with the sum of others. In this way, these two stages are operated iteratively until no more unverified variable nodes in \(C_{\text{unver}}\) can be moved to \(C_{\text{ver}}\) under certain threshold value \(\gamma\) or after maximum \(m\) iterations. In the next iteration, the threshold constraint is eased from \(\gamma\) to \(\gamma+1\). The same procedure repeats again until the threshold value finally reduces to \(\gamma_{\text{max}}\). The VA can be viewed as EVA with \(M = 1\), \(\gamma = 1\) and \(\gamma_{\text{max}} = 1\).

C. Computation Complexity

For the proposed algorithm, the complexity is dominated by the packet-level operations especially when the number of bits in a packet is reasonably large. The packet-level calculations in the EVA include the \(d_v\) XOR calculations in (1) for verification and the \(d_v - 1\) XOR calculations in (2) for correction, where \(d_v\) represents the degree of a check node \(c_i\).

At the first iteration, it is necessary to compute (1) for each check node. After that, we only need to re-compute (1) for the check nodes whose neighboring variable nodes’ current values have been changed in the previous correction stage. Each correction requires the calculation of (2) once and the maximum number of correction operations using (2) is less or equal to the total number of unverified variable nodes after the verification step in the first iteration. As a result, the number of calculations of (2) is less than the codeword length \(n\). Since the total re-computation times of (1) are dependent on the total number of variable nodes corrected in the correction stage, the number of re-computation operations using (1) is very limited when a large percentage of variable nodes can be verified at the verification step in the first iteration. As a result, compared with VA, the extra computation of the proposed algorithm is reasonably low when the channel is in relative good condition.

We note here that in the above analysis, extra process control is required. The extra process control only requires bit-level calculations which are in low complexity and thus has little influence on the total computation load compared with packet-level calculations.

D. Simplified EVA

The optimization of the threshold values is an important issue. In this paper, for EVA, we adopt an “exhaustive” search approach aiming to achieve a minimum FER. In other words, we choose \(S = \{d_v, d_v - 1, \ldots, 1\}\) for regular case and \(S = \{d_v^{\text{max}}, d_v^{\text{max}} - 1, \ldots, 1\}\) for irregular case with \(\gamma_{\text{max}} = d_v^{\text{max}}\).

\[\gamma_v^{\text{max}} \text{ for irregular case with } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for regular case and } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for irregular case with } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for regular case and } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for irregular case with } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for regular case and } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for irregular case with } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for regular case and } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for irregular case with } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for regular case and } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for irregular case with } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for regular case and } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for irregular case with } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for regular case and } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for irregular case with } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for regular case and } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for irregular case with } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for regular case and } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for irregular case with } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for regular case and } \gamma_{\text{max}} = d_v^{\text{max}}.\]

\[\gamma_v \text{ for irregular case with } \gamma_{\text{max}} = d_v^{\text{max}}.\]
that EV A is the best and all EV A-
induces small packet error probability in each frame. This
for both VA and EV A. This is because the small packet size
We can also find that a small packet size gives a small FER
1, EV A gives better performance than VA for all packet sizes.

Fig. 1. FER of VA and EVA with different packet sizes for irregular LDPC
codes over BSC. The solid lines and dashed lines denote the performance of
VA and EVA, respectively.

In each simulation, the maximum iteration number \( m \) is set
to 10. Moreover, the Frame Error Rate (FER) is employed to
indicate the decoding performance.

A. BSC Channel

Binary symmetric channel is one of the simplest noisy
communication channels. The BSC flips each input bit with
probability \( p \).

In the simulations, the packet sizes of irregular packet-based
LDPC codes are from 32 bits to 256 bits. As shown in Fig. 1, EVA gives better performance than VA for all packet sizes.
We can also find that a small packet size gives a small FER
for both VA and EVA. This is because the small packet size
induces small packet error probability in each frame. This
results in the low FER. Compared with VA, EVA outperforms
at least 100 times better when \( p \leq 10^{-3} \) for 32 bits per packet.

Fig. 2 shows FER performance of VA, EVA-2, EVA-4, EVA-
6 and EVA (EVA-15) for packet size of 64 bits. It can be seen
that EVA is the best and all \( \text{EVA-}x \ (x = 2, 4, 6) \) give better
performance than VA, which is at least one order better in
FER when \( p \leq 10^{-3} \). Among the \( \text{EVA-}x \ (x = 2, 4, 6) \), the
probability of false verifications is reduced by increasing
the number of thresholds and thereby achieving smaller FER.

In Fig. 3, the numerical results indicate numbers of average
packet-level operations of VA, EVA-2, EVA-4, EVA-6 and
EVA (EVA-15) for decoding one frame with packet size of
64 bits over BSC. It can be found that the complexity of all
EVA and EVA-\( x \) \( (x = 2, 4, 6) \) is comparable with that of VA,
and their computation loads are increased by less than twice
against VA. Further, different \( \text{EVA-}x \ (x = 2, 4, 6) \) provide
different trade-offs between performance and computation
complexity. For example, EVA-6 is a good choice to replace
EVA since it has the similar performance as EVA but less
complexity.

B. GE Channel

GE channel\(^2\) is a two-state discrete time Markov channel
model. This model includes a good state, in which the prob-
ability of a packet error is zero, and a bad state, in which
a packet is corrupted with a bit error probability \( \epsilon_b \).
The transition probabilities from good state to bad state and from
bad state to good state are \( p \) and \( q \), respectively.

The parameters of GE channel are set as \( p = 0.01 \) and
\( q = 0.1 \). In GE channel simulations, the packet statistic is
dependent on the packet size and the GE channel is allowed
to change its state after each duration of 256 bits.

Fig. 4 shows the decoding performance among the packet
sizes ranging from 32 bits to 128 bits compared with VA. It can
be seen the EVA gives better performance for any packet size.
For \( \epsilon_b \leq 10^{-3} \), EVA is 10 times better than VA in performance
for any packet size.

In Fig. 5, we compare the performance of the irregular
LDPC codes among VA, EVA-2, EVA-4, EVA-6 and EVA
(EVA-15) for packet size of 128 bits. EVA exhibits the best

\(^2\)The quantities of channel parameters of the GE channel can be computed
by observing the packet error statistics of a flat Rayleigh fading channel in
wireless communications [14].
performance and all EVA and EVA-\(x\) \((x = 2, 4, 6)\) give better performance than VA, which is at least one order better in FER when \(\epsilon_b \leq 10^{-3}\). Among the EVA-\(x\) \((x = 2, 4, 6)\), the one with larger threshold has smaller FER values at the same channel condition due to the reduced likelihood of false verifications.

Fig. 6 represents the average packet-level operations of VA, EVA-2, EVA-4, EVA-6 and EVA (EVA-15) for decoding one frame with packet size of 128 bits over the GE channel. From the curves, EVA-\(x\) \((x = 2, 4, 6)\) and EVA require less than double computation load compared with that of VA. It can be found that the normalized ratios do not fall down to one even when \(\epsilon_b\) is large, because of the assumption that the good channel condition is error free and the low transition probability \(p\).

V. CONCLUSION

Packet-based LDPC codes and their decoding algorithms are very promising tools in wireless communications. In this paper, we have applied an enhanced verification-based decoding algorithm for packet-based LDPC codes over wireless channels. The EVA reduces the likelihood of false verification by enhancing the verification condition, which can be used for both regular and irregular LDPC codes. Based on simulations, EVA outperforms the VA in FER results at least 100 times better by increasing no more than twice computation complexity, when channel parameter of BSC is less than \(10^{-3}\) and it gives at least 10 times superior performance against VA with less than 35% complexity increase, if \(\epsilon_b \leq 10^{-3}\) in GE channel.

REFERENCES