Accepted in IEEE International Conference on Image Processing (ICIP 2020), 2020
Authors accepted manuscript. The final published version available at https://doi.org/10.1109/ICIP40778.2020.9190650

ROBUST H.264 VIDEO DECODING USING CRC-BASED SINGLE ERROR CORRECTION AND NON-DESYNCHRONIZING BITS VALIDATION

Vivien Boussard[⋆†] *Firouzeh Golaghazadeh*[⋆] *Stéphane Coulombe*[⋆]
François-Xavier Coudoux[†] *Patrick Corlay*[†]

[⋆] École de technologie supérieure, Dept. of Software and IT Engineering, Montreal, Canada
[†]UPHF, CNRS, Univ. Lille, ISEN, Centrale Lille, UMR 8520 - IEMN, DOAE, Valenciennes, France

ABSTRACT

In this paper, we introduce a novel cyclic redundancy check (CRC)-based single error correction method which we apply to robust H.264 Baseline video decoding. Unlike state-of-the-art methods, the proposed correction algorithm does not require lookup tables as it determines the error location based on binary operations using the computed link layer CRC syndrome. Since multiple errors can lead to the same CRC syndrome as a single error, verification of the corrected packet is performed through a non-desynchronizing bits validation (NDBV), which forwards only compliant packets to the video decoder. Simulations on the H.264 Baseline profile show an average gain of 3.04 dB and 2.36 dB over state-of-the-art spatio-temporal error concealment (STBMA) and NDBV+STBMA reconstruction methods, respectively, at a residual bit error rate of 10^{-6}.

Index Terms— Video transmission, H.264 Baseline, error correction, cyclic redundancy check (CRC), non-desynchronizing bits (NDB)

1. INTRODUCTION

In the context of video content delivery over unreliable channels, packets can either be corrupted or lost during communication. A wide range of solutions are proposed in the literature to deal with such packets, including retransmission, error concealment and error correction. Although retransmissions ensure the integrity of a packet, they are not desirable in low latency applications due to the delay they incur. Corrupted and lost video packets are thus traditionally handled by spatial and/or temporal error concealment (EC) to recover the missing areas of the video [1–6]. EC methods do not differentiate corrupted from lost packets, and systematically discard received corrupted packets, even those that are only slightly corrupted. Moreover, bad reconstructions propagate due to the spatio-temporal predictions used in video compression standards [7]. Therefore, it would be beneficial to extract useful information from corrupted packets. In practice, erroneous packets are discarded in lower layers of the protocol stack by the widely used cyclic redundancy check (CRC) codes [8] at the link layer. CRC codes are traditionally used for error detection purposes only, but some recent works demonstrate the error correction capabilities of such codes. CRC-based error correction can be performed through iterative decoding [9] or using lookup tables [10–13]. To avoid miscorrection of the corrected packet and ensure the usability of the video packet, an additional validation is required. Some works propose error detection based on syntax analysis of the video packets [14, 15]. In [15], the authors introduce the concept of non-desynchronizing bits (NDBs) applied to the H.264 Baseline profile (i.e., bits that can be hit by errors without causing desynchronization of the bit stream). Their analysis culminates in the proposal of two conditions for a packet to be considered as valid. In this paper, this validation is referred to as non-desynchronizing bits validation (NDBV). Moreover, they show that decoding a valid slightly corrupted packet usually results in better video quality than what obtains with EC [14, 16]. Hence, corrupted packets with errors only affecting NDBs could be kept as candidates, making NDBV very useful for filtering packets with several errors. In this paper, we propose a system comprising a novel CRC-based error correction method and NDBV to enhance the decoding robustness of corrupted H.264-coded video packets. The paper is organized as follows. In section 2, we present the proposed approach. We show and analyze the experimental results in section 3. In section 4, we discuss the applicability of the method to other video standards. We conclude this paper and suggest future works in section 5.

This work was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant. Emails: {Vivien.Boussard, Firouzeh.Golaghazadeh}.1@etsmtl.net, Stephane.Coulombe@etsmtl.ca, {Francois-Xavier.Coudoux, Patrick.Corlay}@uphf.fr

2. PROPOSED CRC-BASED ERROR CORRECTION AND NON-DESYNCHRONIZING BITS VALIDATION

In this section, we introduce a novel CRC-based single-error correction algorithm to identify the error position corresponding to the computed CRC syndrome for a given generator polynomial. We then validate the suitability of the solution through NDBV to ensure that the packet corruption was due to a single error rather than many.

2.1. Cyclic redundancy check

CRC codes are widely used for error detection in lower layers of the protocol stack of a wired or wireless transmission. In the context of real-time video transmission, it is used in Ethernet frames [17] at the link layer to decide whether the received packet should be discarded or forwarded to upper layers. The CRC field is computed at the transmitter as the remainder of the long division of the protected data by a generator polynomial $g(x)$. The resulting remainder $r(x)$ is then appended to the packet and sent through the communication. At the receiver side, the long division by the generator polynomial is performed again on the received packet $p_R(x)$ and the appended remainder. At the end of that process, the newly computed remainder is known as the syndrome, denoted $s(x)$. Given the definition of the CRC computation [8], the syndrome can be expressed as follows:

$$s(x) = p_R(x) \bmod g(x) \tag{1}$$

which is equal to zero when no error occurs during the packet's transmission. When an error occurs, the syndrome $s(x)$ will differ from zero. If we consider an error pattern $e(x)$ with non-null coefficients at error positions, we have:

$$s(x) = (p_R(x) + e(x)) \bmod g(x) \tag{2}$$

When a channel exhibits low and uniform BER, packets are more likely to contain a single error than multiple ones. This paper thus focuses on single error correction.

2.2. CRC-based single-error correction

We aim at retrieving, from a non-null syndrome, the corresponding single error position in the packet. Some methods propose storing each single-error position and its corresponding syndrome in a lookup table [10–13]. However, such tables need memory storage, and must be computed prior to the communication and be updated every time the generator polynomial or the maximum length of the protected data change. Using Eq.(2), we can express the error pattern as:

$$e(x) = P(x).g(x) + s(x) \qquad \forall\, P(x) \tag{3}$$

which means that any binary polynomial $P(x)$ with N coefficients (i.e., in GF(2^N) [18]) produces a CRC-compliant error candidate $e(x)$, as the remainder of the modulo operation (the syndrome) remains the same if we add any multiple of $g(x)$. We are searching for single-error patterns, i.e., $e(x)$ with only one non-null coefficient. Most possible $P(x)$ values do not match our criteria since they produce a pattern containing several errors. We propose a novel approach to search only for single-error patterns that match the syndrome, by exploiting knowledge on the generator polynomial and the syndrome. By representing the aforementioned polynomials as binary vectors [18], the approach is to cancel all non-null values of $s(x)$ from LSB to MSB, by successively performing XORs (denoted $\oplus$) with binary-shifted versions of the generator polynomial. From Eq.(3), we see that in doing so, we build

Algorithm 1 Single Error Correction

Input:
- $\mathbf{s}$: the syndrome vector
- n: the degree of the syndrome
- m: the degree of the payload
- $\mathbf{g}$: the vector associated with the generator polynomial used to compute the CRC

Output:
- E_1 the list of valid error patterns for a single bit error

1: $E_1 \leftarrow \{\}$
2: Let $\mathbf{e}$ be a vector of length $m + n$
3: $\mathbf{e} \leftarrow \mathbf{0} \oplus \mathbf{s}$
4: **if** $\text{sum}(\mathbf{e}) = 1$ **then**
5:   Add $\mathbf{e}$ to E_1
6: **end if**
7: **for** $j = 0$ to $m - 1$ **do**
8:   **if** $e_j = 1$ **then**
9:     $\mathbf{e} \leftarrow \mathbf{e} \oplus (\mathbf{g} \ll j)$
10:     **if** $\text{sum}(\mathbf{e}) = 1$ **then**
11:       Add $\mathbf{e}$ to E_1
12:     **end if**
13:   **end if**
14: **end for**
15: Return E_1

$P(x)$ bit-by-bit as we search for an $e(x)$ containing a single error position. If a single error corrupted the packet, we are guaranteed to find such solution. The single-error search is illustrated in Algorithm 1. The main steps are as follows:

3: Initialization of the error vector, denoted $\mathbf{e}$, as a null vector of length $M = m + n$, which corresponds to the cumulative length of the protected data of length m and the CRC field of length n. We replace the n last positions with the syndrome.

4 & 10: We count the non-null values in the error vector $\mathbf{e}$ to obtain the number of errors in the current error pattern. If the sum equals 1, a single-error candidate is identified.

7: We scan each possible position, from LSB (position 0) to MSB (position $m - 1$). We do not consider the n last positions since doing so would take us out of the range of the packet when performing the XOR operation of length n at this position.

8: We successively cancel every non-null e_j value from LSB to MSB by XORing $\mathbf{e}$ with $\mathbf{g}$ shifted at position j. It is important to note that all known generator polynomials have an LSB and an MSB of 1 [8].

Through the properties of Eq.(3), it can be shown that the proposed elimination method will identify all single-error candidates. Depending on the packet size and the computed syndrome (which depends on the actual error pattern), there can be zero, one or several candidates in the set E_1. For example, if the received computed syndrome is $s(x) = x^{15} + x^{10} + x^8 + x^7 + x^6 + x^1 + 1$ when using CRC-

16-CCITT polynomial generator $g(x) = x^{16} + x^{12} + x^5 + 1$, the algorithm outputs a single error pattern at position x^{43}. Note that each generator polynomial is associated with a period (the maximum length for which every single-error position leads to a unique syndrome). Beyond this period, there can be multiple single-error patterns corresponding to the computed syndrome, thus introducing ambiguity caused by the presence of several candidates. Moreover, some multiple-error patterns can produce the same syndrome as a single-bit error. In such a case, the CRC-based single error correction would recognize a single-error pattern and lead to a bad correction. The existence of such cases calls for the integration of a validation step into our solution to prevent miscorrection.

2.3. Validating packet correction

In the H.264 Baseline profile, which is widely used in mobile terminals, sequences are coded using context-adaptive variable length coding (CAVLC). The analysis of CAVLC syntax elements (SEs) [19] in [15] demonstrates that there are bits in some SEs (defined as NDBs) that do not cause any desynchronization on the bit stream when corrupted. Their simulation results reveal that NDBs constitute about one-third of the whole bit stream. Exp-Golomb codewords (EGC) constitute the main example of such bits, and are structured as follows:

$$\underbrace{0\,0\ldots0}_{N\text{ zero-prefix}}\ \mathbf{1}\ \underbrace{X_1\,X_2\ldots X_N}_{N\text{ INFO}}, \qquad X_i \in \{0,1\}, \forall i \in [1,N] \quad (4)$$

The number of zeros in the zero-prefix part informs the decoder on the number of information (INFO) bits to decode right after the leftmost bit set to 1. It is clear that an error on one of the zero-prefix bits will cause immediate desynchronization, as the number of INFO bits will be wrong and propagate as the packet is decoded, while an error in the INFO part (shown with X_i in Eq.(4)), although leading to a different decoded value, does not directly desynchronize the bit stream. Thus, all the INFO bits are possible NDBs. The proposed NDB validation (NDBV) in [15] is a two-check process that determines whether or not a corrupted packet has errors only on NDBs. First, the packet must be decodable: it must have a correct syntax and semantics. Then, the number of decoded macroblocks (MBs) must be correct. If a received erroneous packet meets these two conditions, meaning that the errors were most probably only on NDBs, the corrupted packet is retained (decoded as is). Otherwise, the corrupted packet is dropped and EC is performed. Integrating NDBV into our solution ensures packet integrity after CRC-based single-error correction. Moreover, it allows packets containing several errors to be forwarded to the video decoder, where errors only affect NDBs.

2.4. Proposed decoding system

We propose a system that includes the novel CRC-based single-error correction and NDBV as shown in Figure 1. First, as usual, if the computed syndrome is null (i.e., CRC is good), the packet is reconstructed. If not, our CRC-based

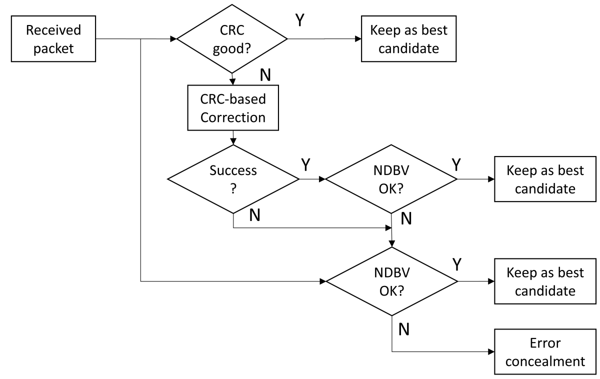
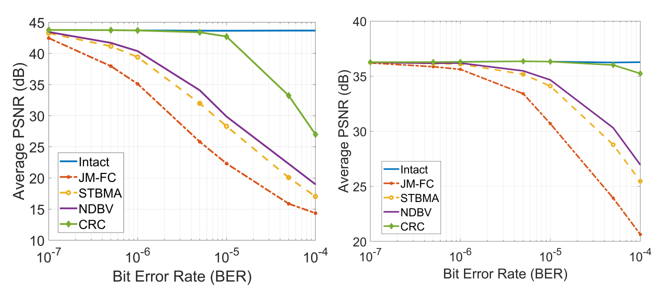


Fig. 1: System of the proposed approach.

single-error search is performed. If a candidate is found at the end of the process, we check the validity of the corrected packet through NDBV. If NDBV fails, we recall the original version of the corrupted packet to perform NDBV once again. This step allows NDB-corrupted packets to be kept as candidates (i.e., reconstructed). If both NDBVs fail, an EC method is performed.

3. SIMULATIONS AND RESULTS

We implemented the proposed system to compare its performance against state-of-the-art methods. The video sequences are coded using the H.264 Baseline profile with the Joint Model (JM) software, version 18.5 [20]. Each packet comprises a slice composed of a single row of MBs. The expected number of MBs in a packet is thus known. We simulate packet corruption for different residual bit error rates (BERs) (i.e., BERs remaining after physical layer error handling) on several video sequences. The BER is applied uniformly and varies from 10^{-7} to 10^{-4} resulting in corrupted packets with zero to several errors. We based our method on the CRC-32 used in the Ethernet protocol [17], as it is widely used and covers the entire packet. Table 1 shows the average PSNR values of the proposed approach versus state-of-the-art methods (JM frame copy (FC) [20], STBMA [2] and NDB [15]), at a BER of 10^{-6} and different Quantization Parameters (QPs), from 22 to 37. In this table, "CRC" denotes the proposed system shown in Figure 1. In our simulations, we



(a) QP = 22 (b) QP = 37

Fig. 2: Average PSNR as a function of the BER for *Ice* sequence at QPs 22 and 37.

(a) JM-FC (19.66 dB) (b) STBMA (23.82 dB) (c) NDBV (24.24 dB) (d) CRC (34.57 dB)

Fig. 3: Visual example of decoding for high BER (10^{-4}) using various approaches on *Ice* at QP=37. Intact is at 36.32 dB.

chose STBMA [2] as the EC method for NDBV and CRC. We observe that the proposed method always outperforms the state-of-the-art, gaining an average of 5 dB in low QPs, as compared to the NDBV method alone, which illustrates the benefit of being able to handle 100% of the single-error cases when the BER is low. We also observe that the proposed method exhibits better performance with increasing video resolution. For a fixed BER, a greater amount of data increases the probability of having an error in a packet. The evolution of the average PSNR when the BER increases is shown in Figure 2, from a BER of 10^{-7} to 10^{-4} on the *Ice* sequence, for two different QPs. The proposed method offers a near-optimal PSNRs at lower BERs, due to the very high probability of having a single error in a corrupted packet. Moreover, the proposed method yields an average PSNR gain of more than 10 dB over other methods from the literature at BER=10^{-5} and QP=22, as illustrated in Figure 2a. The use of NDBV prevents miscorrections for the highest BERs, producing a correction rate at least equivalent to that of the NDBV used alone. A visual example illustrating the performance of the different approaches is shown in Figure 3, for a high BER of 10^{-4} at QP=37. The video has been hit by 31 errors, with 29 single-error and 1 double-error packets. In this example, we gain more than 10 dB, as compared to the NDBV method alone. Such a gain is explained by the proposed method's ability to correct all packets but one, resulting in just one artifact, circled in red in Figure 3.

Table 1: Average PSNR comparison for different sequences and QPs over 100 runs for a BER of 10^{-6}. Simulations were also run over *Harbour*, *Mobcal* and *Park Joy* sequences. Average gain over frame copy (JM-FC) method are shown.

Sequence	QP	**Average PSNR** (dB)				
		Intact	JM-FC [20]	STBMA [2]	NDBV [15]	CRC
Ice 704×576	22	43.69	35.11	39.41 (4.30)	40.37 (5.26)	43.68 (8.57)
	27	41.46	37.30	39.85 (2.55)	40.40 (3.10)	41.46 (4.16)
	32	38.97	37.03	38.19 (1.16)	38.46 (1.43)	38.97 (1.94)
	37	36.30	35.52	36.10 (0.58)	36.19 (0.67)	36.30 (0.78)
City 704×576	22	40.84	29.22	37.19 (7.97)	38.14 (8.92)	40.80 (11.58)
	27	36.65	32.88	36.15 (3.27)	36.25 (3.37)	36.64 (3.77)
	32	33.07	32.12	32.99 (0.87)	32.99 (0.87)	33.07 (0.95)
	37	30.05	29.79	30.02 (0.23)	30.02 (0.23)	30.05 (0.26)
Park Run 1280×720	22	40.41	18.48	26.02 (7.50)	28.88 (10.40)	39.39 (20.91)
	27	35.34	20.12	27.24 (7.12)	29.13 (9.01)	34.93 (14.81)
	32	31.02	22.22	27.61 (5.39)	28.58 (6.36)	30.99 (8.77)
	37	27.39	23.77	26.38 (2.61)	26.66 (2.89)	27.35 (3.58)
Average gain over JM-FC			-	**+3.63**	**+4.38**	**+6.67**
Avg. gain over JM-FC (6 seq.)			-	**+3.37**	**+4.05**	**+6.41**

4. APPLICATION TO OTHER VIDEO STANDARDS

We applied our method to H.264 Baseline video packets as it is the most widely used video standard in mobile communications today. However, the proposed CRC-based single-error correction can be applied to other video standards as it performs error correction at the link layer. Indeed, although HEVC [21] and VVC [22] use context adaptive binary arithmetic coding (CABAC) instead of CAVLC, leading to few NDBs (e.g., most errors will lead to a non-decodable packet, as shown for HEVC in [23]), the two NDBV conditions are still applicable to these standards, as demonstrated in [23] using UDP checksum error correction. Hence, the corrected packet's integrity can still be validated.

5. CONCLUSION AND PERSPECTIVES

In this paper, we proposed a CRC-based error correction system integrating NDBV to ensure the validity of corrected packets. The proposed approach significantly improves the quality of reconstructed H.264 Baseline videos subject to errors and outperforms state-of-the-art approaches. It allows the full recovery of all single-error packets. The addition of NDBV allows forwarding suitable packets containing multiple errors to the decoder. Our simulations show a global increase of the PSNR over all tested sequences, providing a near-optimal quality reconstruction. Future work will include applying the proposed approach to HEVC and VVC compression standards and enhancing the CRC-based error correction method to support the correction of multiple errors.

6. REFERENCES

[1] F. Caron and S. Coulombe, "Video error correction using soft-output and hard-output maximum likelihood decoding applied to H.264 baseline profile," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 7, pp. 1161–1174, 2015.

[2] Y. Chen, Y. Hu, O. C. Au, H. Li, and C. W. Chen, "Video error concealment using spatio-temporal boundary matching and partial differential equation," *IEEE Transactions on Multimedia*, vol. 10, no. 1, pp. 2–15, Jan. 2008.

[3] M. Usman, X. He, M. Xu, and K. Lam, "Survey of error concealment techniques: Research directions and open issues," *IEEE Picture Coding Symposium (PCS)*, pp. 233–238, 2015.

[4] Y. Wang and Q-F. Zhu, "Error control and concealment for video communication - a review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

[5] T. Lin, T. Ding, N. Yang, P. Wu, K. Tung, C. Lai, and T. Chang, "Video motion vector recovery method using decoding partition information," *IEEE Journal of Display Technology*, vol. 12, no. 11, pp. 1451–1463, Nov. 2016.

[6] Z. Zhou, M. Dai, R. Zhao, B. Li, H. Zhong, and Y. Wen, "Video error concealment scheme based on tensor model," *Multimedia Tools and Applications*, pp. 1–17, 2016.

[7] W. T. Tan, B. Shen, A. Patti, and G. Cheung, "Temporal propagation analysis for small errors in a single frame in H.264 video," in *International Conference on Image Processing (ICIP)*, 2008.

[8] J. Sobolewski, *Cyclic Redundancy Check*, John Wiley and Sons Ltd, 2003.

[9] E. Tsimbalo, X. Fafoutis, and R. J. Piechocki, "CRC error correction in IoT applications," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 361–369, Feb. 2017.

[10] S. Shukla and N. W. Bergman, "Single bit error correction implementation in CRC-16 on FPGA," in *International Conference on Field Programmable Technology*. IEEE, Dec. 2004, pp. 319–322, Brisbane, Australia.

[11] S. Babaie, A. K. Zadeh, S. H. Es-Hagi, and N. J. Navimpour, "Double bits error correction using CRC method," in *Fifth international Conference on Semantics, Knowledge and Grid*, Oct. 2009, pp. 254–257.

[12] A. S. Aiswarya and G. Anu, "Fixed latency serial transceiver with single bit error correction on FPGA," in *International Conference on trends in Electronics and Informatics (ICEI)*, May 2017.

[13] X. Liu, S. Wu, X. Xu, J. Jiao, and Q. Zhang, "Improved polar SCL decoding by exploiting the error correction capability of CRC," *IEEE Access*, vol. 7, pp. 7032–7040, 2019.

[14] L. Superiori, O. Nemethova, and M. Rupp, "Performance of a H.264/AVC error detection algorithm based on syntax analysis," *Proceedings of the MoMM*, pp. 49–58, 2006.

[15] F. Golaghazadeh, S. Coulombe, F-X. Coudoux, and P. Corlay, "The impact of H.264 non-desynchronizing bits on visual quality and its application to robust video decoding," in *International Conference on Signal Processing and Communication Systems (ISPCS)*, Dec. 2018.

[16] L. Trudeau, S. Coulombe, and S. Pigeon, "Pixel domain referenceless visual degradation detection and error concealment for mobile video," in *International Conference on Image Processing (ICIP)*, 2011, pp. 2229–2232.

[17] IEEE Standards Association, "IEEE 802.3-2018 - IEEE standard for ethernet," [Online]. Available: https://standards.ieee.org/standard/802_3-2018.html, 2018.

[18] J. Arndt, "Binary polynomials," in *Matters Computational*, pp. 822–863. Springer, Berlin, Heidelberg, 2011.

[19] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[20] "H.264/AVC JM reference software," [Online]. Available: http://iphome.hhi.de/suehring/tml/, version 18.5.

[21] G. J. Sullivan, J-R. Ohm, W-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1648–1667, Dec. 2012.

[22] B. Bross, J. Chen, S. Liu, and Y.-K. Wang, "Versatile video coding (draft 8)," *17th JVET Meeting*, Jan. 2020, Brussels, Belgium.

[23] F. Golaghazadeh, S. Coulombe, F-X. Coudoux, and P. Corlay, "Checksum-filtered list decoding applied to H.264 and H.265 video error correction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, Aug. 2018.