Correcting Localized Deletions Using Guess & Check Codes

ILLINOIS INSTITUTE OF TECHNOLOGY

Serge Kas Hanna and Salim El Rouayheb

Localized Deletions

When deletions occur in a transmitted sequence, the deleted bits are completely removed from the sequence and their positions are unknown at the receiver (unlike erasures). A burst of deletions refers to the case where a certain number of consecutive bits are deleted.

Localized deletions are a more generalized form of bursts of deletions. In this setting, $a \leq b$ deletions are localized within a certain window of length b. These a deletions do not necessarily occur in consecutive positions.

For the problem of correcting a single burst of exactly b deletions, Levenshtein [1] showed that the asymptotic number of redundant bits needed is at least $\log n + b - 1$ bits, where n is the length of the codeword. Schoeny *et al.* [2] derived the same bound non-asymptotically.

The problem of correcting localized deletions arises in several applications. One example is the file synchronization application where a relatively small part of a large file is edited by deleting and inserting characters. Two remote nodes communicate interactively in order to synchronize the localized edits.

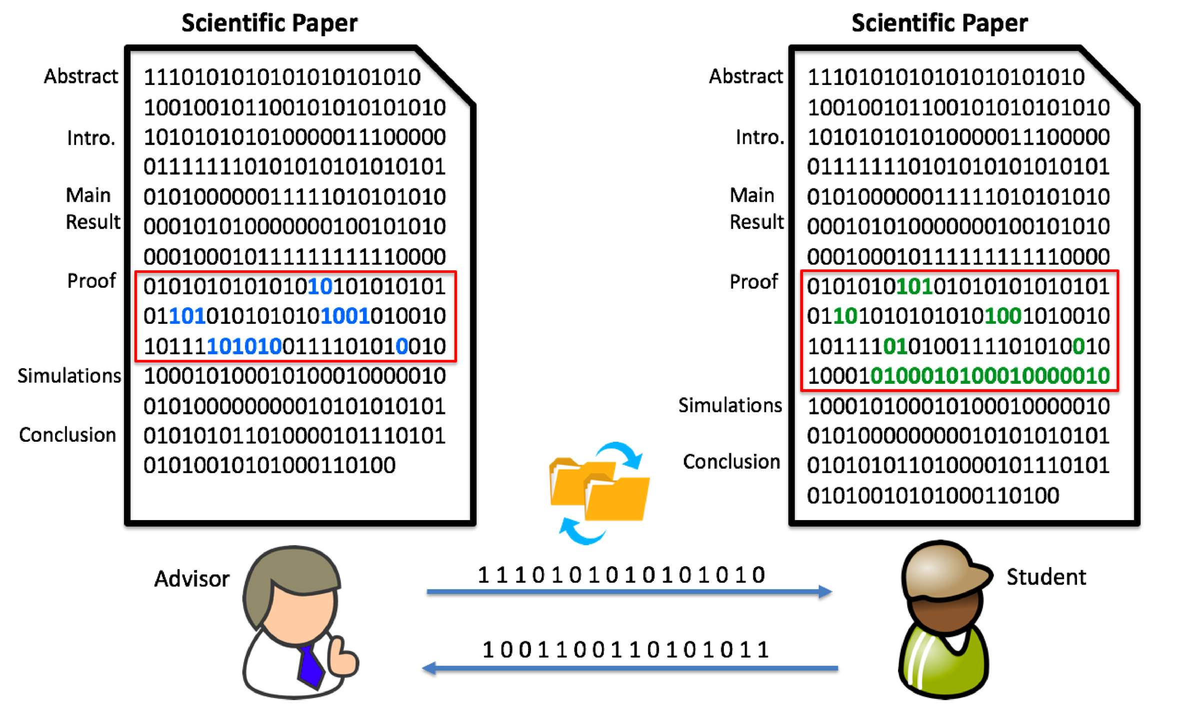

Fig. 1: An example of file synchronization with localized deletions. The student is editing a certain section of a scientific paper, which is shared online with his academic advisor. The two parties communicate interactively in order to synchronize the advisor's version of the paper.

Guess & Check Codes

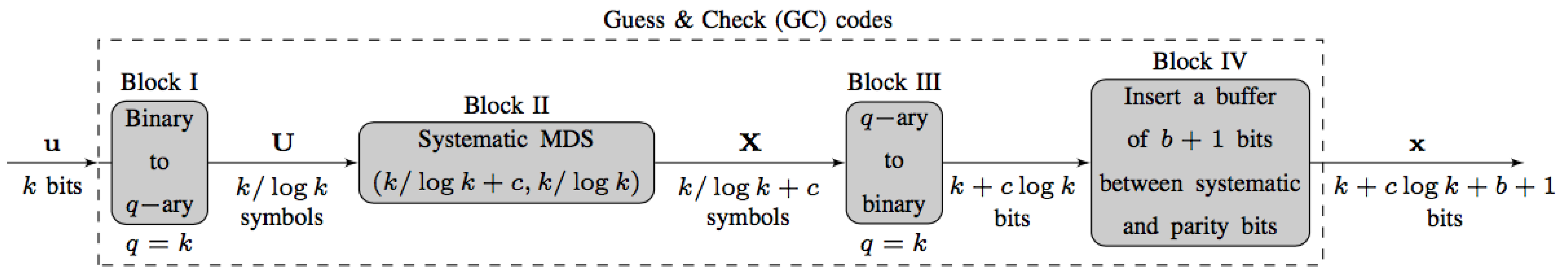

Fig. 2: Encoding block diagram of Guess & Check codes [3] for correcting $a \leq b$ deletions that are localized within a single window $(z = 1)$ of size at most b bits. Block I: The binary message of length k bits is chunked into adjacent blocks of length $\log k$ bits each, and each block is mapped to its corresponding symbol in $GF(q)$ where $q = 2^{\log k} = k$. Block II: The resulting string is coded using a systematic $(k/\log k + c, k/\log k)$ $q-$ary MDS code where c is the number of parity symbols. Block III: The symbols in $GF(q)$ are mapped to their binary representations. Block IV: A buffer of b zeros followed by a single one is inserted between the systematic and the parity bits.

Example: length of message: $k = 16$, length of window: $b = \log k = 4$, field size: $GF(16)$.

1. Encoding (bits in red get deleted):

$$\mathbf{u} = \overbrace{\underbrace{1100}_{\alpha^6}}^{\text{block 1}}\ \overbrace{\underbrace{1010}_{\alpha^9}}^{\text{block 2}}\ \overbrace{\underbrace{0111}_{\alpha^{10}}}^{\text{block 3}}\ \overbrace{\underbrace{1000}_{\alpha^3}}^{\text{block 4}} \longrightarrow \boxed{\text{GC Encoder}} \longrightarrow \mathbf{x} = 1100\ 10{\color{red}10}\ {\color{red}01}11\ 1000\ \overbrace{\mathbf{00001}}^{\text{buffer}}\ \overbrace{1001\ 1000\ 0001}^{\text{parities}}.$$

2. Decoding (3 guesses):

- Guess 1: $\underbrace{11001}_{\mathcal{E}}\ \underbrace{0011}_{\alpha^4}\ \underbrace{1000}_{\alpha^3}\ \overbrace{\underbrace{1001}_{\alpha^{14}}\ \underbrace{1000}_{\alpha^3}\ \underbrace{0001}_{1}}^{\text{parities}} \xrightarrow[\text{first 2 parities}]{\text{Decode using}} \underbrace{0100}_{\alpha^2}\ \underbrace{0110}_{\alpha^5}\ \underbrace{0011}_{\alpha^4}\ \underbrace{1000}_{\alpha^3} \xrightarrow[3^{rd}\text{ parity}]{\text{Check with}}$ ✗
- Guess 2: $\underbrace{1100}_{\alpha^6}\ \underbrace{10011}_{\mathcal{E}}\ \underbrace{1000}_{\alpha^3}\ \underbrace{1001}_{\alpha^{14}}\ \underbrace{1000}_{\alpha^3}\ \underbrace{0001}_{1} \xrightarrow[\text{first 2 parities}]{\text{Decode using}} \underbrace{1100}_{\alpha^6}\ \underbrace{1010}_{\alpha^9}\ \underbrace{0111}_{\alpha^{10}}\ \underbrace{1000}_{\alpha^3} \xrightarrow[3^{rd}\text{ parity}]{\text{Check with}}$ ✓
- Guess 3: $\underbrace{1100}_{\alpha^6}\ \underbrace{1001}_{\alpha^{14}}\ \underbrace{11000}_{\mathcal{E}}\ \underbrace{1001}_{\alpha^{14}}\ \underbrace{1000}_{\alpha^3}\ \underbrace{0001}_{1} \xrightarrow[\text{first 2 parities}]{\text{Decode using}} \underbrace{1100}_{\alpha^6}\ \underbrace{1001}_{\alpha^{14}}\ \underbrace{1100}_{\alpha^6}\ \underbrace{0000}_{0} \xrightarrow[3^{rd}\text{ parity}]{\text{Check with}}$ ✗

Results: Codes for Correcting Localized Deletions

Theorem 1 (Code properties for correcting one set of localized deletions) *Guess & Check (GC) codes can correct in polynomial time $a \leq b$ deletions that are localized within a single window of size at most b bits, where $m \log k + 1 \leq b \leq (m+1)\log k + 1$ for some constant integer $m \geq 0$. Let $c > m + 2$ be a constant integer. The code has the following properties:*

1. *Redundancy: $n - k = c\log k + b + 1$ bits.*
2. *Encoding complexity is $\mathcal{O}(k \log k)$, and decoding complexity is $\mathcal{O}\left(k^3/\log k\right)$.*
3. *Probability of decoding failure:*

$$Pr(F) \leq \frac{k^{m+4}}{k^c \log k} - (m+2)\frac{k^{m+3}}{k^c}. \tag{1}$$

Theorem 2 (Code properties for correcting $z > 1$ sets of localized deletions) *Guess & Check (GC) codes can correct in polynomial time $z > 1$ sets of $a \leq b$ deletions, with each set being localized within a window of size at most b bits, where $m\log k + 1 \leq b \leq (m+1)\log k + 1$ for some constant integer $m \geq 0$. Let $c > z(m+2)$ be a constant integer. The code has the following properties:*

1. *Redundancy: $n - k = zc\log k + z^2 b + z$ bits.*
2. *Encoding complexity is $\mathcal{O}(k \log k)$, and decoding complexity is $\mathcal{O}\left(k^{z+2}/\log^z k\right)$.*
3. *Probability of decoding failure:*

$$Pr(F) = \mathcal{O}\left(\frac{k^{z(m+4)}}{k^c \log^z k}\right). \tag{2}$$

Numerical Results: Simulations on the Probability of Decoding Failure

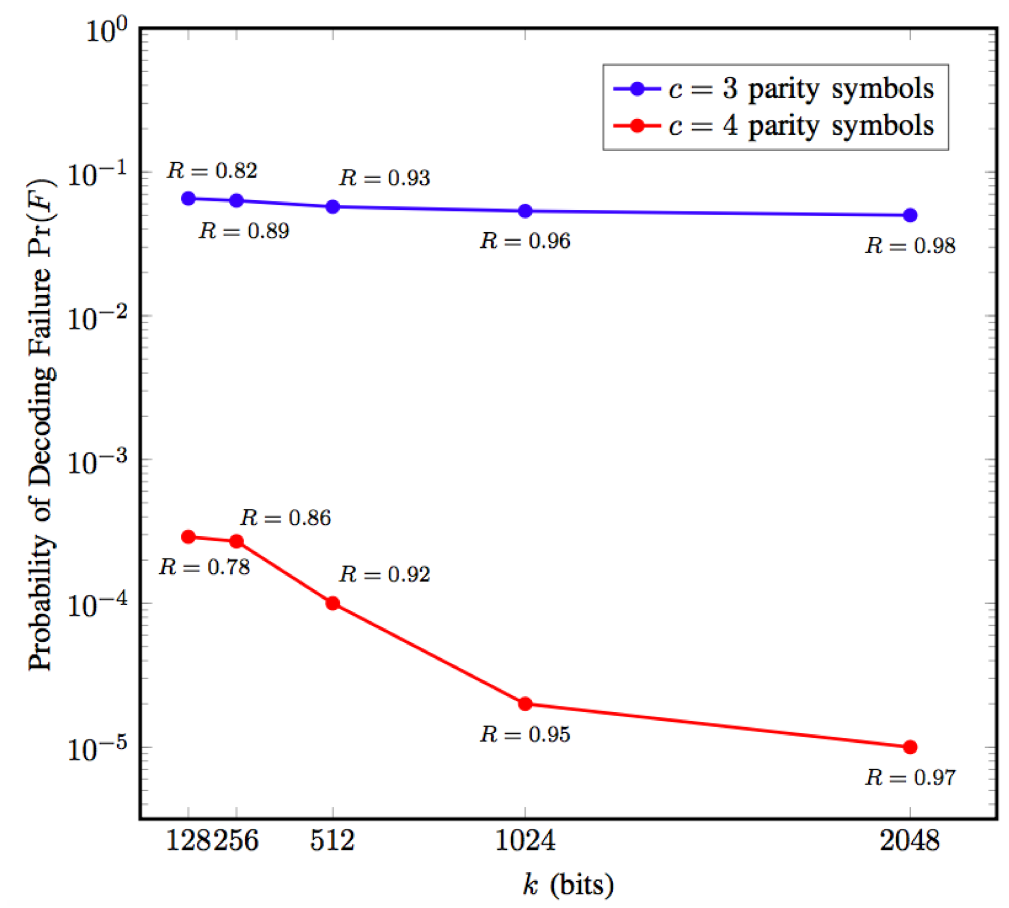

Fig. 3: ($a = b = \log k$ localized deletions) The graph shows the probability of decoding failure $Pr(F)$ of GC codes for different message lengths k. The results of $Pr(F)$ are averaged over 10000 runs of simulations. The window position in which the deletions are localized is also chosen uniformly at random.

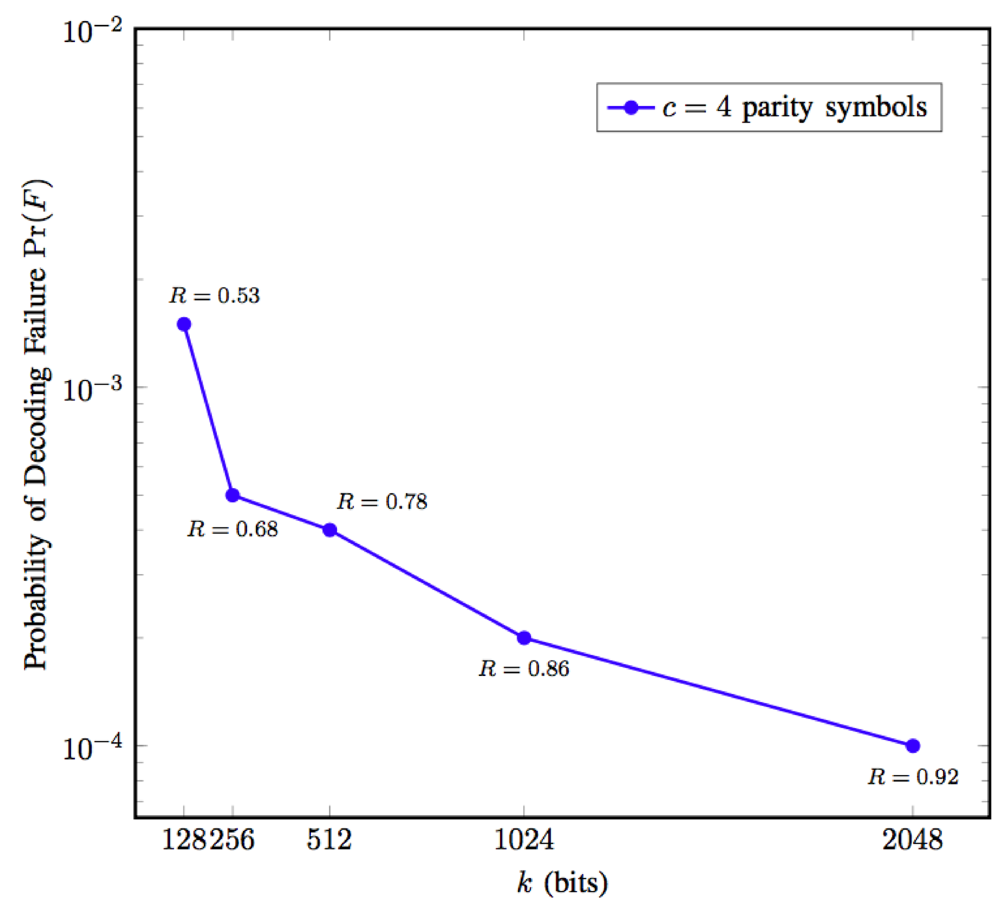

Fig. 4: ($\delta = 3$ non-consecutive deletions) The graph shows the probability of decoding failure $Pr(F)$ of GC codes for different message lengths k. The results of $Pr(F)$ are averaged over 10000 runs of simulations. The positions of the deletions is chosen uniformly at random.

References

[1] V. Levenshtein, "Asymptotically optimum binary code with correction for losses of one or two adjacent bits," *Problemy Kibernetiki*, vol. 19, pp. 293-298, 1967.

[2] C. Shoeny, A. Wachter-Zeh, R. Gabrys and E. Yaakobi, "Codes correcting a burst of deletions and insertions," *IEEE Transactions on Information Theory*, vol. 63, pp. 1971-1985, April 2017.

[3] S. Kas Hanna and S. El Rouayheb, "Guess & Check Codes for Deletions, Insertions, and Synchronization" submitted to *IEEE Transactions on Information Theory*, 2017.