Rank–Deficit Hybrid Adaptive Fulcrum Decoder for Low–Complexity Network Coding

Sovanjyoti Giri*, Satya Prakash†
Department of Electronics and Communication Engineering,
National Institute of Technology Tiruchirappalli, India.
(Email: sovanjyoti@nitt.edu*, 208124027@nitt.edu†)

***Abstract*—Random Linear Network Coding (RLNC) achieves robustness against packet loss but requires dense decoding over large Galois fields. Fulcrum network codes lower the decoding complexity by combining an inner $\mathrm{GF}(2)$ linear code with an outer $\mathrm{GF}(2^h)$ linear code that does not compromise on the reliability at the receivers. This paper proposes a *Rank-Deficit Hybrid Adaptive Fulcrum* (RDHAF) decoder that exploits the rank-deficit, calculated after the inner decoding operation. Essentially, RDHAF decoder first performs Gaussian elimination for inner decoding, computes the rank deficit ρ with respect to the source packets, and then lifts only the ρ-dimensional residual subsystem for outer decoding. Using the classical flop counts for Gaussian elimination and a weighted cost model, we prove that RDHAF never exceeds the asymptotic complexity of the existing standard advanced adaptive (AA) Fulcrum decoder. Simulation results over packet-erasure channels with typical Fulcrum parameters confirm that RDHAF indeed reduces the computational cost of the decoding operation in comparison to AA decoder while maintaining the same decoding probability.**

***Index Terms*—Computational cost, Decoding complexity, Fulcrum network coding, Gaussian elimination, Rank-deficit, RLNC.**

I. Introduction

In the context of beyond 5G wireless research, which is increasingly focusing on requirements such as ubiquitous Mobile Broadband and enhanced Ultra-Reliable Low-Latency Communication, network coding [1] offers improvement in transmission efficiency by allowing intermediate nodes to mix data streams during packet forwarding. Linear Network Coding (LNC) is often preferred due to its structural simplicity, while Random Linear Network Coding (RLNC) relies on coding coefficients randomly drawn from finite fields to maintain robustness under changing network conditions [2]. In generation-based RLNC [3], source packets are encoded within a block (defined as a generation), and decoding is possible only after all coded packets from that generation are received. Sliding window techniques encode smaller packet groups that are transmitted incrementally [4], allowing decoding to begin earlier. To improve decoding delay, systematic transmission can be employed, where uncoded packets are sent first and coded packets are used only as redundancy [5].

The choice of finite field size in RLNC directly affects the encoding and decoding performances and the related computational effort [2]. Lower-order fields reduce computational complexity but increase the likelihood of linear dependence among coded packets, which degrades decoding performance. Higher-order fields mitigate this issue but incur higher complexity. Fulcrum Network Coding [6] addresses a trade-off by applying outer encoding in a higher-order Galois field followed by a mapping to a lower-order field in inner encoding. With this structure, Fulcrum network codes operate as an RLNC-based FEC scheme that was initially proposed to address the heterogeneity in 5G communication and storage systems [7].

For Fulcrum codes, [6] presents different decoder designs based on receiver capabilities, distinguishing between high-end receivers with strong computational resources and low-end receivers with limited processing capabilities. To improve decoding probability with reduced complexity, [8] introduces an adaptive decoding algorithm that adds a fixed number of redundant packets; however, this may generate linearly dependent packets, reducing transmission efficiency and decoding probability. To further improve the decoding probability, an advanced adaptive decoding (AA) algorithm is proposed in [9] that attempts inner decoding first and invokes outer decoding upon failure, although it applies outer decoding over the entire set. In some other work, [10] proposes the idea of tunable expansion packets, where the number of expansion packets is adjusted dynamically to meet a target decoding probability.

In view of the existing literature, here we discuss about two key observations on Fulcrum network decoders. Firstly, once the $\mathrm{GF}(2)$ elimination in inner decoding is finished, the decoder has enough information to determine which among the first n columns are independent and which are not yet resolved (i.e., the corresponding original packets have not yet been recovered) by inner elimination. Therefore, the n-dimensional outer system is more computationally complex than required when only a small subset of the original packets remains undecoded. Secondly, the existing complexity analyses for Fulcrum decoding are largely simulation-based. Although [6], [8], [9] mention operation counts in terms of finite-field arithmetic and matrix operations for decoding, there is no unified analytical framework for exploring the asymptotic computational complexity of different decoding strategies.

Motivated by the existing gaps in the literature as described above, in this paper, we introduce the *Rank-Deficit Hybrid Adaptive Fulcrum* (RDHAF) decoder and summarize the key contributions as the following:

- The proposed RDHAF scheme is a hybrid decoding scheme that retains the standard Fulcrum encoding while introducing a modified decoding strategy . Essentially, the Fulcrum decoder exploits the residual rank deficiency identified after inner decoding over $\mathrm{GF}(2)$ to reduce the number of operations for outer decoding over $\mathrm{GF}\left(2^h\right)$.
- Using the standard flop counts for Gaussian elimination [11], we explore the computational complexities of

the standard AA and RDHAF decoders, and prove that RDHAF never exceeds the asymptotic complexity of AA. In fact, complexity of RDHAF is strictly lower under typical operating conditions (smaller expansion packets and moderately dense inner matrix).
- We run Monte-Carlo simulations to compare the performance of RDHAF with existing standard decoding methods for different erasure rates and delay budgets. Results indicate, RDHAF noticeably reduces the computational cost of decoding without compromising performance.

II. System Model

A. Fulcrum Encoder

We adopt the Fulcrum network coding framework presented in [6], [8], [9]. Following the generation based fulcrum encoding, each generation consists of n source packets as $\boldsymbol{p} = [p_1, \ldots, p_n]^\top$. An outer encoder expands them to $(n+r)$ intermediate symbols over $\mathrm{GF}(2^h)$:

$$\boldsymbol{u} = \mathbf{C}\boldsymbol{p} \tag{1}$$

where $\mathbf{C}$ is an $(n+r)\times n$ matrix over $\mathrm{GF}(2^h)$ with $h > 1$. In line with the Fulcrum network coding framework, $\mathbf{C}$ can be interpreted as a systematic precoding matrix [6], where the first n rows correspond to the original packets and the remaining r rows generate expansion packets through linear combinations over $\mathrm{GF}(2^h)$. Without any loss of generality we assume that $\mathbf{C}$ is fixed and known to the receiver.

After outer encoding, the i-th coded packet is formed through inner encoding as a random combination of the $(n+r)$ intermediate symbols over $\mathrm{GF}\,(2)$ as the following,

$$x_i = \boldsymbol{\lambda}_i \boldsymbol{u} \tag{2}$$

Next, x_i is sent to the receiver along with its coefficient vector $\boldsymbol{\lambda}_i$ (of $1\times(n+r)$ dimension) in the header. Here one can note that, although the intermediate symbols $\mathbf{u}$'s are defined over $\mathrm{GF}(2^h)$, the inner coding coefficients $\boldsymbol{\lambda}_i$'s belong to $\mathrm{GF}(2)$. This interaction is well defined because $\mathrm{GF}(2^h)$ forms a vector space over $\mathrm{GF}(2)$, so $\mathrm{GF}(2)$ coefficients act as valid scalars on $\mathrm{GF}(2^h)$ elements. Consequently, linear combinations formed using $\mathrm{GF}(2)$ coefficients remain valid $\mathrm{GF}(2^h)$ symbols.

B. Channel Model

Packets are transmitted over a discrete-time, memoryless, wireless erasure channel characterized by Bernoulli packet loss with erasure probability ε. Let m be the number of packets successfully received by the receiver.

C. Decoding Strategies

For m received packets at the receiver, let us form the received packets vector $\boldsymbol{y}$ as $[y_1, \ldots, y_m]^\top$. Stacking the coefficient vectors of the received packets yields the inner decoding matrix $\mathbf{A}_{\mathrm{in}}$ that is an $m\times(n+r)$ matrix over $\mathrm{GF}(2)$.

Inner Decoder (ID): It performs Gaussian elimination on $\mathbf{A}_{in}$ over $\mathrm{GF}(2)$. Decoding succeeds only if $\mathrm{rank}_{\mathrm{GF}(2)}(\mathbf{A}_{in}) \geq n + r$ (where $\mathrm{rank}_{\mathrm{GF}(2)}(\mathbf{A}_{\mathrm{in}})$ denotes the number of linearly independent rows of $\mathbf{A}_{\mathrm{in}}$ over $\mathrm{GF}(2)$). Here, if the inner decoding is successful, the receiver recovers all intermediate symbols and hence the original packets $\boldsymbol{p}$ using the known outer transform w.r.t. precoding matrix $\mathbf{C}$.

Outer Decoder (OD): First of all, it maps each row of $\mathbf{A}_{in}$ to $\mathrm{GF}(2^h)$. Specifically, during outer decoding, each coefficient row of $\mathbf{A}_{\mathrm{in}}$ is mapped to a coefficient row in $\mathrm{GF}(2^h)$ by multiplying the same with the outer precoding matrix $\mathbf{C}$. Stacking all such rows yields the $m \times n$ outer decoding matrix $\mathbf{A}_{out}$ over $\mathrm{GF}(2^h)$. Next, OD solves the linear system $\mathbf{A}_{out}\boldsymbol{p} = \tilde{\boldsymbol{y}}$ using Gaussian elimination to get back $\boldsymbol{p}$ (where $\tilde{\boldsymbol{y}}$ denotes the modified received vector obtained by applying the same row operations, as found during inner decoding, on $\boldsymbol{y}$). The outer decoding succeeds only if $\mathrm{rank}_{\mathrm{GF}(2^h)}(\mathbf{A}_{out}) = n$.

Advanced Adaptive (AA) Decoder: As described in [9], AA decoding happens as the following,

- if $m < n + r$, outer decoding is invoked directly;
- if $m \geq n + r$, AA first attempts inner decoding. If $\mathrm{rank}_{\mathrm{GF}(2)}(\mathbf{A}_{in}) \geq n + r$, then decoding succeeds; otherwise AA falls back to outer decoding.

III. Proposed RDHAF Decoder

In this paper, we propose the RDHAF (Rank-Deficit Hybrid Adaptive Fulcrum) decoding method that takes advantage of the fact that inner Gaussian elimination over $\mathrm{GF}(2)$ identifies those components of the received symbol vectors which are associated with the rank deficiency of the inner decoding matrix. Instead of performing computationally expensive operations on $\mathrm{GF}(2^h)$ (during outer decoding) throughout the received symbol set, RDHAF selectively restricts the high-complexity operations only to the rank-deficient components indicated by the inner decoder, thus achieving significant reduction in overall decoding complexity. The term *hybrid* is used in the name RDHAF due to the fact that the encoding part is kept fully identical to the standard Fulcrum technique retaining its interoperability and structural benefits, while the decoding part is redesigned to be adaptive and rank-aware.

A. Rank-Deficiency Calculation

First, we write $\mathbf{A}_{in}$ as, $\mathbf{A}_{in} = [\mathbf{A}_1 \ \mathbf{A}_2]$ where $\mathbf{A}_1$ is an $m \times n$ matrix over $\mathrm{GF}(2)$ corresponding to the first n columns (original symbols w.r.t. uncoded packets), and $\mathbf{A}_2$ is an $m \times r$ matrix over $\mathrm{GF}(2)$ corresponding to the expansion packets. During inner decoding, RDHAF performs Gaussian elimination on $\mathbf{A}_{in}$ to obtain a row-echelon form (REF),

$$\mathbf{R} = \mathbf{U}\mathbf{A}_{in} \tag{3}$$

where $\mathbf{U}$ is an $m\times m$ invertible binary matrix representing the row operations. Let, $\mathcal{P} \subseteq \{1, \ldots, n + r\}$ be the set of pivot columns of R, and we define,

$$K = |\mathcal{P} \cap \{1, \ldots, n\}|, \qquad \rho = n - K \tag{4}$$

Here, one can infer that, ρ is the number of non-pivot columns in the original symbol submatrix $\mathbf{A}_1$, and it is denoting the *rank-deficiency* while performing the inner decoding.

B. RDHAF Decoding Algorithm

RDHAF decoder is built upon the *rank-deficiency* characterization as given in Section III-A. After obtaining the matrices $\mathbf{R}$ and $\mathbf{U}$ as in (3), the decoder proceeds as the following.

1) **Inner decoding and rank-deficiency calculation**: Using the row-echelon matrix $\mathbf{R}$, the rank deficiency ρ with respect to the inner decoding is computed as given in (4). The value of ρ determines whether additional outer decoding is required or not.
2) **Case distinction for outer decoding**
   - **Case $\rho = 0$ (no rank deficiency):** All original symbols correspond to pivot columns in $\mathbf{R}$. In this case, the source packets $\boldsymbol{p}$ are fully recovered through the inner $\mathrm{GF}(2)$ decoding stage alone, and the RDHAF decoder terminates without going to Step 3.
   - **Case $\rho > 0$ (rank deficiency present):** The ρ non-pivot columns among the first n positions of $\mathbf{R}$ are identified. The same row operations $\mathbf{U}$ obtained during inner elimination are then applied to the corresponding rows of the linear system at OD, resulting in a reduced $m \times \rho$ residual matrix $\mathbf{B}_\rho$ that contains only the remaining unsolved components of $\boldsymbol{p}$.
3) **Outer decoding**
   Gaussian elimination is performed on $\mathbf{B}_\rho$ over $\mathrm{GF}(2^h)$ to recover the ρ unsolved source packets; remaining $(n-\rho)$ packets are directly obtained from inner decoding stage.

IV. Computational Complexity Analysis

Here, we adopt the classical operation counts for Gaussian elimination on an $a \times b$ linear system [11]. The number of arithmetic operations is expressed as,

$$\Phi(a,b) = \Theta(ab^2 + b^3), \tag{5}$$

which becomes $O(ab^2 + b^3)$ in Big–O notation.

For $\mathrm{GF}(2)$, addition and subtraction operations are of unit cost. For fields like $\mathrm{GF}(2^h)$, with $h > 1$, multiplications are significantly more expensive than additions. Following these arguments and the weighted–cost model of [6], [9], we write the computational complexities of Gaussian elimination in $\mathrm{GF}(2)$ and $\mathrm{GF}(2^h)$ as the following respectively,

$$\Phi_{\mathrm{GF}(2)}(a,b) = \Phi(a,b) \tag{6}$$

$$\Phi_{\mathrm{GF}(2^h)}(a,b) = \gamma\,\Phi(a,b) \tag{7}$$

where $\gamma(>1)$ denotes the relative computational cost of operation over $\mathrm{GF}(2^h)$, compared to $\mathrm{GF}(2)$. Here, the constant factors in row-operation counts are omitted.

A. Computational Complexity of the AA Decoder

For each generation, the AA decoder first performs an inner $\mathrm{GF}(2)$ elimination on the $m \times (n+r)$ matrix $\mathbf{A}_{\text{in}}$, imparting the computational cost as the following (using (5) and (6)),

$$C_{IAA}(m, n+r) = O\big(m(n+r)^2 + (n+r)^3\big) \tag{8}$$

Whenever the inner step does not provide a complete solution, AA falls back to an outer $\mathrm{GF}(2^h)$ elimination on an $m \times n$ system with computational cost (using (5) and (7)),

$$C_{OAA}(m,n) = O(mn^2 + n^3) \tag{9}$$

Now, AA decoder either imparts the computational cost as $C_{IAA}(m, n+r)$ if the inner decoder succeeds, or the computational cost as $C_{IAA}(m, n+r) + C_{OAA}(m,n)$ if the inner decoder fails. Consequently, we get the best and worst-case computational complexities respectively as,

$$C_{AAB} = O\big(m(n+r)^2 + (n+r)^3\big) \tag{10}$$

$$C_{AAW} = O\big((m+n+r)(n+r)^2 + (m+n)n^2\big) \tag{11}$$

B. Computational Complexity of the RDHAF Decoder

RDHAF performs the same inner elimination on $\mathbf{A}_{\text{in}}$ as in AA decoder. So, the inner decoder computational cost in this case $\big($denoted as $C_{IRD}(m, n+r)\ \big)$ is same as $C_{IAA}(m, n+r)$. Now, when $\rho = 0$, the first n columns are already full rank and RDHAF uses only inner elimination. On the other hand, when $\rho > 0$, RDHAF constructs an $m \times \rho$ residual system over $\mathrm{GF}(2^h)$ and performs Gaussian elimination only on this subsystem. Hence, the corresponding outer decoding computational cost with rank-deficit ρ is $\big($using (5) and (7)$\big)$,

$$C_{\text{ORD}}(m,\rho) = \Phi_{\mathrm{GF}(2^h)}(m,\rho) = \gamma\,\Theta(m\rho^2 + \rho^3) \tag{12}$$

Now, let us consider a random variable R denoting the rank-deficit. If the expected rank-deficit is $\bar{\rho} = \mathbb{E}[R]$, the expected computational complexity of RDHAF can be represented as,

$$C_{RD} = C_{IRD}(m, n+r) + \gamma\,\mathbb{E}\big[\Phi(m,R)\big] \tag{13}$$

Since $0 \le R \le n$, we get, $R^2 \le nR$ and $R^3 \le n^2R$. Now, monotonicity of expectation (Th. 2.34, [12]) and (5) yield,

$$\begin{aligned}\mathbb{E}[\Phi(m,R)] &\le c\big(m\,\mathbb{E}[R^2] + \mathbb{E}[R^3]\big)\\ &\le c\big(mn\,\mathbb{E}[R] + n^2\mathbb{E}[R]\big) = c(mn+n^2)\bar{\rho}\end{aligned} \tag{14}$$

for some constant $c > 0$. Next, substituting $C_{IRD}(m, n+r)$ and $\mathbb{E}[\Phi(m,R)]$ into (13) yields,

$$C_{RD} = O\big(m(n+r)^2 + (n+r)^3 + (mn+n^2)\bar{\rho}\big) \tag{15}$$

C. Complexity Comparison of AA and RDHAF Decoders

Case 1 (Inner decoder succeeds): This case corresponds to the scenario where the rank of the inner decoding matrix is not less than $n+r$. In this scenario, both AA an RDHAF decoders stop after the inner decoding matrix elimination (corresponds to $\rho = 0$ case in RDHAF), So, the computational complexity of the decoding techniques are same, and in reference to Section IV-A and IV-B, we write that,

$$C_{RD} = C_{AA} = O\big(m(n+r)^2 + (n+r)^3\big) \tag{16}$$

Case 2 (Inner decoder fails): In this scenario, AA goes for the outer matrix elimination with the aim of searching for the full n dimensions, whereas RDHAF goes for the outer elimination in search of the residual ρ dimensions (where $0 < \rho < n$). Now, with reference to Section IV-A and IV-B, the complexity of AA is given in (11), whereas the same for RDHAF will be the same as in (15) with the difference that $0 < \bar{\rho} < n$ here (as $0 < \rho < n$ for any ρ). Since, $(mn+n^2)\bar{\rho} < (mn+n^2)n = (m+n)n^2$ for $0 < \bar{\rho} < n$, from the expressions of (11) and (15), we deduce that,

$$C_{RD} < C_{AA} \tag{17}$$

In view of (16) and (17), it is realized that the computational complexity of RDHAF decoding is no worse than the same of AA under any circumstance. In fact, under random linear fulcrum encoding with erasure channel consideration, the inner decoder will fail with high probability, and Case 2 is highly likely to occur. Consequently, one can infer that the computational burden due to RDHAF decoder is likely to be lesser than the same of AA in most of the scenarios.

V. Performance Analysis

We use Monte-Carlo simulation to evaluate the performance of our proposed RDHAF decoder with respect to the relevant, standard existing schemes given below:

- Fulcrum Inner Decoder (ID),
- Fulcrum Outer Decoder (OD),
- RLNC–GF(2) Decoder (RL2),
- RLNC–GF(2^8) Decoder (RL256),
- Normal Adaptive Decoder[1] (NA),
- Advanced Adaptive Decoder (AA).

Regarding the simulation setup, considering generation-based network coding, we consider a generation of size $n = 16$ and the number of expansion packets is $r = 2$. These assumptions are consistent with the prior works [6], [8], [9]. Also, in all simulation setup, we transmit $S = (n + \delta)$ packets[2] and record the number of received packets m with erasure probability ε. Regarding the coding approaches, the inner coding vectors are generated with Bernoulli entries with parameter 0.5, and RLNC over GF(2) uses the same density. RLNC over GF(2^8) picks random coefficients with uniform distribution, and for the outer decoder also we choose the same field GF(2^8). All the approaches share the same outer precoder matrix $\mathbf{C}$ (wherever needed). For each decoding scheme, the reported simulation results in the following subsections are obtained by averaging over $10{,}000$ independent trials.

The computational cost metric used in Sections V-A and V-B is defined as $\mathcal{A}_{tot} = A_{\text{add}} + 2.4A_{mult}$, where A_{add} and A_{mult} denote the total number of finite-field additions and multiplications, respectively. Here, the weighting factor 2.4 is to account the higher computational cost of multiplication operations in a fair complexity comparison with the addition operations. Specifically, the value 2.4 is selected based on the experimentally measured ratio between the execution times of finite-field multiplication and addition as reported in [9].

A. Decoding Computational Cost versus Delay

Fig. 1 shows the computational cost versus delay, $\delta = S-n$, for a fixed erasure probability $\varepsilon = 0.2$. As the delay increases, more packets are likely to be available at the decoder, and all schemes initially exhibit a rise in computational cost due to the larger matrices being processed. Beyond a moderate delay ($\delta \approx 5$–6 in Fig. 1), the cost of the adaptive schemes become saturated and then starts decreasing as the additional packets allow the decoders to avoid costly outer decoding. Although, both NA and AA decoders get benefited from additional packets, NA has lower computational cost than AA because it directly selects the decoding mode based on the number of received packets. In contrast, AA always attempts ID first and invokes OD only upon failure, which introduces additional inner elimination operations and increases the overall cost.

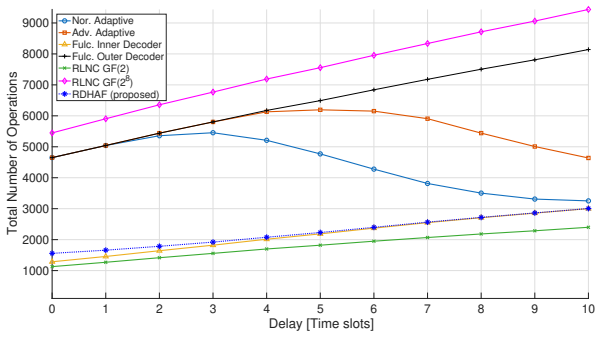

Fig. 1. Computational Cost vs. Delay (δ) for $\varepsilon = 0.2$.

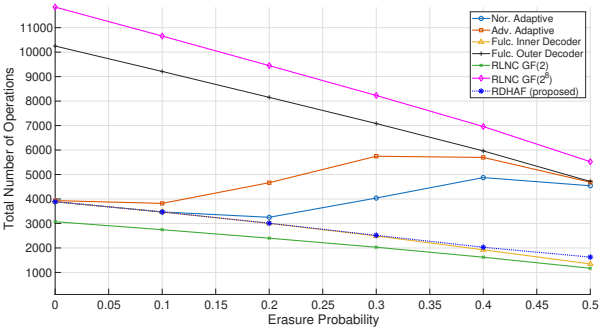

Fig. 2. Computational Cost vs. Erasure Probability for $\delta = 10$.

In Fig. 1, RL256 has the highest cost since it always operates over GF(2^8). Fulcrum OD also exhibits high cost because it performs outer decoding over GF(2^8) whenever possible. Here, the proposed RDHAF decoder not only consistently achieves the lowest cost among OD, NA, AA, and RL256, but also shows very close performance to Fulcrum ID.

B. Decoding Computational Cost versus Erasure Probability

Fig. 2 depicts the computational cost comparison with respect to erasure probability ε for a fixed deadline $\delta = 10$. For all schemes, the cost decreases as ε approaches 0.5 since fewer packets are available before the deadline and the size of the linear system shrinks.

Here, NA and AA exhibit non-monotonic behavior for the following reason. For small ε, the inner decoder often succeeds and the cost is close to the same of ID; as ε increases, the inner attempt fails more frequently, triggering more use of outer decoder, increasing the overall cost. Next, for high erasure probabilities, enough packets are not collected to justify an n-dimensional elimination and the cost decreases again. Throughout Fig. 2, RDHAF tracks the lower envelope of the AA curve. When inner decoder succeeds, both AA and RDHAF have similar cost; when it fails, RDHAF performs only a ρ-dimensional outer elimination while AA performs an n-dimensional one. As a result, RDHAF remains significantly

[1]NA decoding selects the decoding type at the receiver, based on the number of packets available at the decoding instant and a decoding priority. The same enables a trade-off between decoding probability and computational effort.

[2]Here, r denotes the number of expansion packets generated by the outer encoder and is fixed for a given generation [10], whereas δ represents the additional transmission allowance beyond the generation size [8], [9].

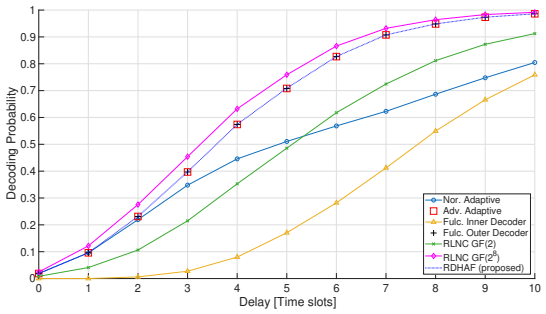

Fig. 3. Decoding Probability vs. Delay (δ) for $\varepsilon = 0.2$.

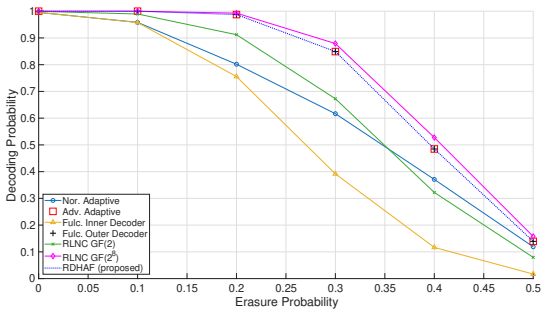

Fig. 4. Decoding Probability vs. Erasure Probability for $\delta = 10$.

lower than AA and Fulcrum OD, while RL256 remains the most expensive scheme. Although, Fulcrum ID and RL2 have the lowest computational costs, we will see in Section V-C that they offer the lowest decoding probabilities.

C. Decoding Probability Analysis

Fig. 3 depicts the decoding probability graphs versus delay δ for a fixed erasure rate $\varepsilon = 0.2$. As delay increases, receivers gather more packets, and hence all schemes gradually approach a decoding probability of 1. Here, RL256 curve converges the earliest due to large-field operations, followed closely by Fulcrum OD, AA, and the proposed RDHAF decoder. Fulcrum ID and RL2 exhibit slower convergence because decoding in $\mathrm{GF}(2)$ suffers from linear dependencies between the coefficient vectors. Between NA and AA, AA achieves faster convergence because it attempts ID after each packet arrival and invokes OD immediately upon ID failure, whereas NA bases its decision solely on the number of received packets, which may delay the OD and consequently the overall decoding process. Next, the crossover between RL2 and NA curves at a delay nearly at $\delta = 5$ happens due to the fact that RL2 continues to benefit from the increase in linear independence as more packets arrive, whereas the NA decoder may invoke OD once ID fails, even when subsequent packet arrivals could further improve the linear independence.

Fig. 4 further examines the decoding performance with respect to ε for $\delta = 10$. With the increase of erasure probability, fewer packets reach the decoder at the deadline, causing performance degradation in all schemes. RL256 again shows the best performance, while the performance of Fulcrum ID and RL2 degrade sharply due to limited rank growth.

The results of Fig. 3 and 4 convey that RDHAF achieves nearly identical decoding performance to AA and Fulcrum outer decoder. This confirms that restricting $\mathrm{GF}(2^8)$ operations to ρ-dimensional subspace *does not compromise* the rank recovery and consequently the decoding performance.

Remarks: Regarding the simulations, the erasure probability is considered in the range of 0 to 0.5 to capture the behaviour of a realistic lossy channel. With the increase of the erasure probability beyond 0.5, the probability of collecting sufficient number of linearly independent coded packets at the receiver within a realistic delay constraint becomes lesser and lesser, that consequently lowers the decoding probability significantly for almost all schemes under consideration. Eventually, the simulation results will provide limited comparative insight.

VI. Conclusion and Future Work

This paper has presented the RDHAF decoder for Fulcrum Network Codes that exploits the rank-deficit in the inner decoding elimination. Using the standard Gaussian-elimination flop counts, we have shown that RDHAF never exceeds the asymptotic complexity of AA. Performance evaluation of RDHAF confirms that it achieves a consistent reduction in computational cost relative to AA without any compromise in the decoding performance. Next, our future work focuses on characterizing its decoding statistics that will complement the computational complexity analysis and help in quantifying a tradeoff between complexity and reliability.

References

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on information theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] Y. Li, E. Soljanin, and P. Spasojevic, "Effects of the generation size and overlap on throughput and complexity in randomized linear network coding," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 1111–1123, 2011.

[3] S. Giri and R. Roy, "Performance analysis of feedback-based network-coded systems for broadcast," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2544–2560, 2021.

[4] F. Wu, Z. Tan, H. Zhu, and P. Dong, "Reliability and latency analysis of sliding network coding with re-transmission," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2023, pp. 1–6.

[5] I. Chatzigeorgiou and A. Tassi, "Decoding delay performance of random linear network coding for broadcast," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7050–7060, 2017.

[6] D. E. Lucani, M. V. Pedersen, D. Ruano, C. W. Sørensen, F. H. Fitzek, J. Heide, O. Geil, V. Nguyen, and M. Reisslein, "Fulcrum: Flexible network coding for heterogeneous devices," *Ieee Access*, vol. 6, pp. 77 890–77 910, 2018.

[7] D. E. Lucani, M. V. Pedersen, J. Heide, and F. H. Fitzek, "Coping with the upcoming heterogeneity in 5g communications and storage using fulcrum network codes," in *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*. IEEE, 2014, pp. 997–1001.

[8] V. Nguyen, J. A. Cabrera, G. T. Nguyen, F. Gabriel, C. Lehmann, S. Mudriievskyi, and F. H. Fitzek, "Adaptive decoding for fulcrum codes," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2018, pp. 133–139.

[9] V. Nguyen, J. A. Cabrera, D. You, H. Salah, G. T. Nguyen, and F. H. Fitzek, "Advanced adaptive decoder using fulcrum network codes," *IEEE Access*, vol. 7, pp. 141 648–141 661, 2019.

[10] V. Nguyen, E. Tasdemir, G. T. Nguyen, D. E. Lucani, and F. H. Fitzek, "Tunable expansion packets for fulcrum codes," in *European Wireless 2019; 25th European Wireless Conference*. VDE, 2019, pp. 1–7.

[11] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.

[12] D. P. Kroese and Z. Botev, *An Advanced Course in Probability and Stochastic Processes*. Chapman and Hall/CRC, 2023.