On Optimization of Wireless XOR Erasure Codes

Jalaluddin Qureshi¹, Adeel Malik²

¹Department of Computer Science, National University of Computer & Emerging Sciences, Karachi, Pakistan
²Communications & Networking (CNN) Laboratory, Dankook University, Yongin-si, South Korea

Abstract—In this paper we study the problem of optimizing linear erasure codes over $GF(2)$ for single-hop wireless multicasting. We first present an algorithmic optimization technique which minimizes the number of transmissions given the knowledge of packets received by each of the receivers. We show that the algorithmic optimization technique is equivalent to an NP-complete Boolean constraint satisfaction problem (CSP), using Schaefer’s dichotomy theorem. This makes the performance evaluation of such transmission network a difficult problem. We derive closed form expression and upper bound of minimum number of transmissions for restricted classes of the problem which sheds interesting insight about the behavior of optimal erasure codes over $GF(2)$. We also show that the greedy algorithm on the mapped Boolean SAT problem outperforms previously proposed state-of-the art heuristic coding algorithms, and its performance is virtually similar to the optimal maximum distance separable (MDS) code for network parameters of practical interest.

Keywords: Code Optimization; Wireless Multicasting; Network Coding; Efficient Algorithms;

I. INTRODUCTION

Wireless multicasting is an efficient method of disseminating common data to multiple receivers. There has been a dramatic increase in multicast traffic load in mobile cellular networks, and it is projected that such growth will continue in 5G networks [2]. The wireless channel however is susceptible to packet erasures due to interference and burst errors due to signal fading. A packet may also be considered corrupted when the number of bit errors in a packet exceeds the maximum number of bits which can be corrected by error correction code (ECC) adopted by the transmitter.

It has been shown that instead of using the automatic repeat request (ARQ) protocol of retransmitting an erased or corrupted packet, adopting forward error correction (FEC) at packet level known as erasure coding (also referred as network coding, a broader coding technique [25, Chapter 4], in many related research works) improves the transmission throughput [14].

Erasure coding is a class of forward error correction (FEC) which corrects packet erasures and burst errors. Instead of retransmitting packets, erased packets are coded before transmission. Consider for example an access point (AP) multicasting packets $P_1$ and $P_2$ to receivers $R_1$ and $R_2$. Receiver $R_1$ did not receive $P_1$, and $R_2$ did not receive $P_2$. Instead of individually retransmitting these two packets, $P_1$ and $P_2$ can be XOR coded, and the coded packet $P_1 \oplus P_2$ is transmitted which both $R_1$ and $R_2$ can decode to recover their erased packets.

Erasure coding has been proposed as an efficient method to improve reliability for single-hop wireless multicast applications due to the broadcast nature of wireless transmission, such as in wireless sensor networks (WSNs) to disseminate software updates for debugging and task modifications [7], and multicasting over Wi-Fi and cellular networks [9], [2]. It has also been adopted in various transmission standards. The Raptor R10 erasure code for instance has been adopted in third generation partnership project (3GPP) multimedia broadcast multicast service (MBMS) for multicasting file delivery and streaming applications, and the Reed-Solomon (RS) erasure code is used in digital video broadcasting (DVB) for multicasting live video [1].

The significance of single-hop wireless multicast services is evident from the fact that the IEEE 802.11aa Task Group recently standardized a set of protocols to support reliable audio video (AV) multicast over wireless local area networks (WLAN). Single-hop wireless multicast data traffic will also increase in 5G networks due to growing popularity of mobile videos, and in wireless sensor network (WSN) and internet of things (IoT) due to frequent software upgrades [2], [18].

There is special interest in erasure codes over $GF(2)$ as it involves the relatively simple operation of XOR addition for encoding and decoding. This minimizes the computation cost during encoding and decoding, and complexity of hardware implementation. Computational complexity has been identified as one of the most challenging issues in 5G system with power-constrained devices [2]. Implementation of random linear network code (RLNC) on TmoteSky sensor node has shown that decoding coded packets over $GF(2)$ is at least 6.5 times faster than decoding coded packets over $GF(2^8)$ [18]. It is due to these reasons that the widely adopted Raptor R10 code, Luby-Transform (LT) code, erasure coding for energy and memory constrained WSN and low-end IoT devices is constructed over $GF(2)$ [7], [18].

Given the popularity of wireless multicasting and $GF(2)$ erasure code, a significant research work exists on the optimization of $GF(2)$ erasure codes to minimize the number of transmissions under various assumptions [19], [22], [7], [18], [24], [20], [21], [13], [11], [23]. Despite the popularity of $GF(2)$ erasure code for wireless multicast, non-asymptotic research results on the fundamental properties of such code is limited. To address this gap we derive closed form expression on the exact minimum number of transmissions and upper
bound on the minimum number of transmissions for restricted classes of this problem. These results shed interesting insight about the behavior of optimal $GF(2)$ erasure codes.

To the best of our knowledge this is the first work of its kind, where an optimization algorithmic technique has been proposed to minimize the total number of transmissions for wireless multicasting using $GF(2)$ erasure code. By presenting a mapping scheme which maps any instance of the optimization problem to a restricted subclass of Boolean SAT problem, our work opens avenues to study more efficient algorithms for the problem considered in the paper, as the Boolean SAT problem is a canonical NP-complete problem with a wealth of efficient heuristic SAT solver. We further show that a greedy algorithm on the optimization algorithmic technique performs virtually similar to the optimal maximum distance separable (MDS) code for network parameters of practical interest. Part of the work presented in this paper appears in [16].

The rest of the paper is organized as follows. Preliminaries on related bibliography and system model is given in Section II. Algorithmic optimization technique and proof of its intractability is presented in Section III. Evaluation of efficient heuristic algorithms is presented in Section IV. Based on the results of rank distribution to generate independent packet for three receivers in Section V, derivation of closed form expression and upper bound for restricted classes of the problem is presented in Section VI. We conclude with the main results of our paper in Section VII.

II. PRELIMINARIES

When the finite field size $q$ is given as $n \leq q$, where $n$ is the number of receivers in the network, it has been shown that a linearly independent packet can be constructed for all the receivers in polynomial time based on the Sanders-Li algorithm [6], such a code is also known as MDS code. The performance evaluation of MDS code presented in [10], serves as the lower bound of the minimum number of transmissions for any linear erasure code, as each client need to receive exactly $k$ MDS coded packets before it can decode the $k$ input packets. For $q=2$, the optimization algorithm and throughput performance for $n=2$ is known, hence in this paper we consider $n \geq 3$.

The problem which we study in the paper is closely related with the index coding problem (ICP), which is a transmission minimization problem by a transmitter having a set of $k$ input packets $P = \{P_1, P_2, ..., P_k\}$, to a set of $n$ receivers, $\mathcal{R} = \{R_1, R_2, ..., R_n\}$. The transmitter has the side information knowledge of the set of packets each of the receiver has in its cache, known as the has set $H_i \subseteq P$, and the set of packets each of the receiver wants, known as the want set $W_i \subseteq P \setminus H_i$.

It has been shown that the ICP is an NP-Hard problem in general [19]. The problem has also been shown to be computationally intractable for the subclasses where each receiver wants unique packets (unicast) [15], when multiple receivers want the same packet (multicast) [5], and when all the receivers want all the packets, i.e. $W_i \cup H_i = P, \forall R_i$ (multicast) [22], [7], [6], [8]. All these results assume linear coding over $GF(2)$, and memoryless decoding.

In memoryless decoding a receiver can only buffer input packets, and a coded packet which is not instantly decodable is dropped. Decoding is performed using the substitution method, in which a coded packet generated by XOR of $w$ input packets is decoded using XOR addition if the receiver has any of the $w-1$ packets used to generate the coded packet.

The results of our paper are also compared with RL code over $GF(2)$. In an RL code, the coding coefficients are randomly and uniformly selected from $GF(q)$ [17].

A. System Model and Notation

Consider an access point (AP) multicasting $k \geq 2$ input packets $P = \{P_1, P_2, ..., P_k\}$ to $n \geq 3$ receivers $\mathcal{R} = \{R_1, R_2, ..., R_n\}$. The AP can transmit and generate coded packets by linearly coding input packets over $GF(2)$, and a receiver can buffer input and coded packets. Packet erasure at each receiver is assumed to be independent and identically distributed (iid), following the Bernoulli model with packet loss probability of $p$, $0 \leq p < 1$, and successful packet reception probability of $s = 1-p$.

The expected number of transmissions before all receivers have received $k$ linearly independent packets is denoted by $E[t]$, and the retransmission rate $R_t$ is defined as the ratio $E[t]/k$, i.e. the expected number of retransmissions for each input packet before all receivers are satisfied.

Similar to the assumption used in previous ICP papers [6], [7], [8], [12], [14], [19], [20], [22], [23] to simplify analysis, we assume that there exists a reliable feedback channel for the transmitter to receive ACKs from the receivers.

Multi-user feedback for single hop wireless multicasting has also been practically implemented on testbed. Ferreira et al. have implemented a multi-user feedback mechanism for coded Wi-Fi multicasting in which batch based Acknowledgment (ACK) transmission scheme and channel statistics are used to estimate packet losses at different receivers [9]. Dong et al. implement packet reception status from neighboring nodes using request vectors on TelosB wireless sensor network, in which the receiver informs the transmitter of the packet reception status of $k$ packet in a batch [7].

The matrix $M_i, M_i \in GF(2)^{r_i \times k}$, represents the coding coefficient matrix of the innovative packets receiver $R_i$ has received. The rank of the matrix $M_i$ is denoted by $r_i$, $r_i \leq k$. We call a packet innovative if it is linearly independent with respect to previously received packets. Each row of $M_i$ represents the coding vector of a linearly independent packet $R_i$ has received.

The coding vector of an input packet $P_j$ is represented by the standard basis vector $e_j$. Any packet which is found to be linearly dependent, after performing triangularization step of Gaussian elimination, is dropped. Once receiver $R_i$ has received $k$ innovative packets, it can decode the $k$ input packets by the operation $M_i^{-1}C_t$, where $C_t$ is the vector of $k$ packets. Matrix inversion $M_i^{-1}$ is performed using Gaussian elimination or back-substitution if $M_i$ is a triangular matrix. We call a receiver $R_i$ satisfied iff $r_i = k$, and unsatisfied otherwise.
We exclude the zero coding vector from any consideration in our paper.

Our optimization problem is not a restricted class of the ICP, as we consider that \( H_i \subset \text{span}(\mathcal{P}) \), i.e. the receiver can buffer coded packets in addition to the input packets. Further in our case, the decoder can decode the coded packets using both substitution method (instant decoding) and matrix inversion.

**Problem Statement** GIP\(_2\): Given the knowledge of input and coded packets each of the unsatisfied receiver has \( H_i \subset \text{span}(\mathcal{P}) \), generate a coded packet over \( GF(2) \) which will be innovative for all the receivers (Generate Independent Packet Lemma 1.

We now present the following result.

Consider as an example the following instance of the ICP multicast problem under the restriction of memoryless decoding is different. Their mapping scheme assumes that the decoder can decode using only the substitution method [7, Definition 1], [8, Constraint 3], [22, Equations (3)-(4)].

The optimal solution (minimum number of transmissions) of the ICP multicast problem is denoted by \( \text{OPT}_1 \). Similarly the optimal solution of the GIP\(_2\) problem is denoted by \( \text{OPT}_2 \). We now present the following result.

**Lemma 1.** \( \text{OPT}_1 \geq \text{OPT}_2 \)

**Proof.** We first consider instances in which the optimal code can be decoded using the substitution method. As decoding in ICP multicast problem and GIP\(_2\) problem can be done using substitution method, the optimal solution for both the problems will be equal. This proves the equality relationship.

We next consider instances in which the optimal code can only be decoded using the matrix inversion method. As decoding cannot be done using matrix inversion in the ICP multicast problem, additional transmissions would be needed to decode the packets using the substitution method for the ICP multicast problem. This proves the strict inequality relationship.

Combining the results of these two relationships we hence complete the proof.

**Example 1:** Consider as an example the following instance of the problem for \( n = 6 \), \( k = 3 \), \( H_1 = \{P_1, P_2\} \), \( H_2 = \{P_1, P_3\} \), \( H_3 = \{P_2, P_3\} \), \( H_4 = \{P_1\} \), \( H_5 = \{P_2\} \), \( H_6 = \{P_3\} \). For the ICP multicast problem, one can verify using the exhaustive search approach that after the transmission of any linear coded packet over \( GF(2) \), there exists at least one receiver with \( r_i = 1 \). Clearly to satisfy the receiver with \( r_i = 1 \), at least two additional transmissions are required, hence \( \text{OPT}_1 \geq 3 \).

However in the GIP\(_2\) problem, as the receiver can buffer innovative packets which cannot be instantly decoded, by transmitting \( P_1 \oplus P_2 \) and \( P_2 \oplus P_3 \), all receivers can decode their requested packets using the substitution method and matrix inversion, hence \( \text{OPT}_1 \geq 3 \geq \text{OPT}_2 = 2 \).

**III. Optimization Algorithm**

In this section we present an optimization algorithm for the GIP\(_2\) problem. Our formulation is based on the fact that for a matrix \( M_i \), the rank of its rows is equal to the rank of its columns. The coding vector we wish to construct is denoted by \( w = [x_1, x_2, \ldots, x_r] \), \( x_j \in GF(2) \). Append \( w \) as the last row of all matrices \( M_i \) which are not full rank matrices. After appending \( w \), the size of \( M_i \) will be given as \( (r_i + 1) \times k \).

For each matrix \( M_i \) perform the triangulization step of Gaussian elimination on the columns. This will then generate \( k - r_i \) columns with elements in its first \( r_i \) rows equal to zero. The elements in the last row will be given by some linear combination of the variables \( x_j \).

The problem of generating an innovative vector \( w \) for a receiver is now equivalent to assigning Boolean values to \( x_j \) such that at least one column with first \( r_i \) elements equal to zero, has the \( r_i + 1^{th} \) element equal to one. This is represented by the disjunction of the linear combination (XOR addition) of \( x_j \) variables in the elements of the \( r_i + 1^{th} \) row, for columns with first \( r_i \) elements equal to zero. We denote this by the propositional formula \( y_i \). As the rank of the matrix is \( r_i \), there will be \( k - r_i \) such linear combination equations.

To solve the problem GIP\(_2\) the Boolean assignment to \( x_j \) should be such that \( w \) is innovative for all \( M_i \), i.e. all the propositional formula \( y_i \) should be true. We illustrate our optimization algorithm with the aid of an example.

**Example 2:** Consider as an example the following instance of the GIP\(_2\) for \( n = 5 \), \( k = 4 \), \( H_1 = \{P_1, P_4, P_3\} \), \( H_2 = \{P_3, P_4\} \), \( H_3 = \{P_1, P_2, P_3, P_4\} \), \( H_4 = \{P_1, P_2, P_3\} \) and \( H_5 = \{P_1 \oplus P_4, P_2, P_3\} \). After performing triangulization on the columns of \( M_i \), the following propositional formulas are generated.

\[
y_1 = x_2 \lor (x_1 \lor x_4 = 1), \quad y_2 = x_1 \lor x_2, \\
y_3 = (x_2 \lor x_3 = 1) \lor (x_2 \lor x_4 = 1), \\
y_4 = x_4, \quad y_5 = (x_1 \lor x_4 = 1).
\]

We illustrate the triangulization step on matrix \( M_1 \).

\[
\begin{pmatrix}
1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0
\end{pmatrix}
\rightarrow
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0
\end{pmatrix}
\]

If the conjunction of \( y_i \), i.e. \( y_1 \land \ldots \land y_5 \), is satisfiable then the problem GIP\(_2\) can be solved for the given instance. Note that \( y_1 = x_2 \lor (x_1 \lor x_4 = 1) \), because the first \( r_2 = 2 \) elements in the second and fourth column of \( M_1 \) are zero.

**A. Hardness of GIP\(_2\)**

In this section we evaluate the computational hardness of solving arbitrary instances of the problem GIP\(_2\). We first show that the GIP\(_2\) is equivalent to the Boolean constraint satisfaction problem (CSP), and then show that the Boolean CSP is an NP-complete problem using Schaefer's dichotomy theorem. The restriction in our propositional formula of GIP\(_2\) makes it a subclass of the Boolean SAT problem, which is a subclass of the Boolean CSP.
Lemma 2. The GIP$_2$ problem is equivalent to the Boolean CSP, with the restriction that the propositional formula $y_i$ is given by the disjunction of non-negation variables and XOR of two or more non-negation variables.

Proof. We show that the Boolean algebra can be used to transform the propositional logic $y_1 \land \ldots \land y_k$ into a conjunctive normal form (CNF). A CNF is given by conjunction of clauses, and each clause is a disjunction of literals, and each literal is either a non-negation Boolean variable $x_i$, or its negation $\overline{x}_i$.

XOR of two variables can be decomposed into CNF using the following relationship,

$$x_{j1} \oplus x_{j2} = (x_{j1} \lor x_{j2}) \land (\overline{x}_{j1} \lor \overline{x}_{j2}).$$

Similarly XOR of three or more variables can also be decomposed into CNF using Boolean algebra. An illustration of the decomposition of XOR of 3 variables is given as follow,

$$x_{j1} \oplus x_{j2} \oplus x_{j3} = (x_{j1} \lor x_{j2} \lor x_{j3}) \land (x_{j1} \lor \overline{x}_{j2} \lor \overline{x}_{j3}) \land (\overline{x}_{j1} \lor x_{j2} \lor \overline{x}_{j3}) \land (\overline{x}_{j1} \lor \overline{x}_{j2} \lor x_{j3}).$$

As $\lor$ is distributive over $\land$, disjunction of a literal with a propositional logic given by XOR can be transformed into CNF. We illustrate such decomposition for disjunction of a variable with XOR of two variables.

$$(x_{j1} \lor x_{j2}) \lor x_{j3} = ((x_{j1} \lor x_{j2}) \land (\overline{x}_{j1} \lor \overline{x}_{j2})) \lor x_{j3} = (x_{j1} \lor x_{j2} \lor x_{j3}) \land (\overline{x}_{j1} \lor \overline{x}_{j2} \lor x_{j3}).$$

The restriction on the class of satisfiability problem is given as follow. Each propositional formula $y_i$ is given by the disjunction of non-negation variables and XOR of two or more non-negation variables. Based on the transformation techniques above, each formula $y_i$ can be transformed in to CNF, and hence the propositional logic $y_1 \land \ldots \land y_k$ will also be a CNF. This completes the proof. \hfill \square

Lemma 3. The Boolean CSP with the restriction stated in Lemma 2 is an NP-Complete problem.

Proof. According to Schaefer’s dichotomy theorem, a Boolean CSP is either an NP-complete problem or a tractable problem. The Boolean CSP is tractable if it has any one of the following six properties, 0-valid, 1-valid, Horn formula, dual-Horn formula, 2-CNF or Affine-SAT, otherwise it is an NP-Complete problem [4, Chapter 12]. We show that with the restriction stated in Lemma 2, the Boolean CSP does not satisfy any one of these six properties.

0-valid, 1-valid: A CNF is 0-valid (or 1-valid) if all clauses can be satisfied by assigning zero (or one) to all variables. This happens when at least one non-negation variable (or one negation variable) exists in each clause. It is easy to verify that this property is not satisfied as $y_i$ can be given by only XOR addition such as $x_{j1} \oplus x_{j2} = 1$, in which case $y_i$ will be satisfiable iff $x_{j1}$ and $x_{j2}$ are assigned different Boolean values.

Horn and dual-Horn formula: A CNF is a Horn formula (or dual-Horn) if every clause has at most one non-negation (or negation) variable. Based on the transformation shown in the proof of Lemma 2 it is easy to verify that a clause can contain more than one non-negation (or negation) variables.

2-CNF: A CNF is 2-CNF, if all clauses contains at most two literals. This property is not satisfied when $M_i$ has three or more dependent columns.

Affine-SAT: A CNF is an Affine-SAT if all clauses can be written as $x_{j1} \oplus \ldots \oplus x_{jm} = \omega$, $\omega \in \{0, 1\}$. This property is not satisfied when $y_i$ is given by disjunction of non-negation variables and XOR of two or more variables.

As the general Boolean CSP with restriction shown in Lemma 2 does not satisfy any one of the six properties, it is hence an NP-complete problem. This completes the proof. \hfill \square

Theorem 1. The GIP$_2$ problem is an NP-Complete problem.

Proof. It is easy to verify that the GIP$_2$ problem belongs to NP, as given an assignment of the variables $x_j$ it can be verified in polynomial time whether or not GIP$_2$ is satisfiable. As the GIP$_2$ problem belongs to NP and is equivalent to the Boolean CSP as shown in Lemma 2 which has been shown to be an NP-Complete problem as shown in Lemma 3, hence the GIP$_2$ problem is an NP-Complete (NPC) problem. This completes the proof. \hfill \square

IV. EFFICIENT ENCODING

As the GIP$_2$ problem is an NPC problem, in this section we evaluate the performance of a greedy approach on the optimization algorithm proposed in Section III, and compare its performance with recently proposed state of the art coding schemes in which encoding decision is made based on packet reception status, namely the Vertex-Coloring Dynamic [8], Weight-Pick encoding [12] and Hamming-D [23] schemes.

A. Proposed Greedy Scheme

For our proposed greedy scheme, we first construct propositional formulas $y_i$ based on the method discussed in the previous section. For each matrix $M_i$, with the appended vector $w$, after Gaussian elimination each of the $\gamma_i^{th}$ element, $\gamma \leq k$, in the last row is given by some linear combination (i.e. XOR addition) of the variables $x_j$. We are only interested in those columns where the first $r_i$ elements are equal to zero (there are $k-r_i$ such columns). The linear combination in the last element of such column should be equal to 1 for the vector $w$ to be linearly independent.

We use a $(k-r_i) \times (k+1)$ matrix $Q_i$, where each row stores the variables $x_j$ present in the linear combination of an element of interest in the last row (the $r_i+1^{th}$ row) of $M_i$. The elements in matrix $Q_i$ are saved as follow, if variable $x_j$ is present in the linear combination of the equation in element $a_{r_i+1,g}$ of $M_i$, then element $b_{r,j}$ of $Q_i$ will be 1, and 0 otherwise.

The elements in the $k+1^{th}$ column of $Q_i$ save the value (0 or 1) which the XOR addition of variables represented in the $\gamma_i^{th}$ row of $Q_i$ should be equal to, so that the linear combination is satisfied. This is initialized to 1.

A $k \times 1$ matrix $L_i$ is used to count the number of unassigned variables $x_j$ present in the $\gamma_i^{th}$ row of $Q_i$. If only one unassigned variable $x_j$ is present in the $\gamma_i^{th}$ row of $Q_i$, and
scheme performs close to the optimal encoding scheme of MDS code.

### B. Simulation Results

The results of our simulations are plotted in Figure 1. For the Hamming-D algorithm [23], in our simulation we assume that the encoder can select arbitrary number of packets. In accordance with [12] we adopt $C_{max} = 5$ for Weight-Pick algorithm, which is the maximum number of packets which can be encoded by the encoding scheme.

The results of the graphs show that a greedy scheme on our proposed optimization algorithm performs significantly better than the previous state of the art erasure coding scheme over $GF(2)$ for wireless multicasting. Further the performance of the greedy assignment scheme is virtually similar to the MDS code. The MDS code are “optimal” code, as the receiver need coded packets before it can decode the $k$ matrix $T_i$.

Using the matrices $Q_i$, $L_i$, $S_i$ and $T_i$ for each of the $n$ receivers, a greedy approach to find assignment of 0 and 1 to $x_j$ such that the maximum number of receivers receive an innovative packet is given as a pseudo-code in Table I. Before the start of the greedy algorithm none of the variables $x_j$ are assigned any values of 0 or 1, and will be called as unassigned variables.

For each of the maximum $k$ unassigned variables $x_j$ we first temporarily assign it a value of 1, and then evaluate how many propositional formulas $y_i$ will be satisfied. A propositional formula will be satisfied if the linear combination given as $y_i = x_j$ exist, determining whether such a linear combination exist, only the $c_j$ element of $S_i$ needs to be read. The same procedure is then repeated for the temporary assignment of 0 to each of the $x_j$ variables, and evaluating whether $y_i = (x_j = 0)$. This is done by reading the $d_j$ element of $T_i$. Resulting in a maximum of $2 \cdot k$ evaluations.

Based on these $2 \cdot k$ evaluations, we then (permanently) assign 0 or 1 to an unassigned variable which satisfies the maximum number of propositional formulas. After the assignment of value to $x_j$, this value is then XOR added in the element $b_{\gamma,k+1}$ of $Q_i$. The element $b_{\gamma,j}$ is assigned 0 as $x_j$ has been assigned a value. And the matrices $L_i$, $S_i$ and $T_i$ are updated accordingly.

The same procedure of evaluating the assignment of values to the remaining unassigned variables is repeated. The repetition procedure continues until either all receivers are satisfied or all variables have been assigned a value.

### C. Computational Complexity of the Proposed Scheme

The computational complexity of Gaussian elimination on $n$ matrices $M_i$ is given as $O(nk^3)$. The computational complexity of initializing the elements of matrix $Q_i$ to 0 for $n$ matrices is given as $O(nk^2)$. Similarly as $L_i$ counts the number of 1’s in each row of $Q_i$, the complexity of generating such $L_i$ matrices is $O(nk^2)$. Matrices $S_i$ and $T_i$ can be generated with worst case complexity of $O(nk^2)$. Hence the computational
complexity of generating the matrices \( Q_i, L_i, S_i \) and \( T_i \) is \( O(nk^2) \).

The computational complexity of the pseudocode in Table I is given as follows. The complexity of step 1 is given as \( O(nk) \) due to the two for loops, step 1 will be repeated for a maximum of \( k \) times due to the while loop, resulting in total complexity of \( O(nk^2) \).

In step 2 of the pseudocode, if the condition that \( c_t \) is equal to one will be true for a maximum of \( nk \) times as there are \( n \) matrices \( Q_i \), each with \( k \) rows. If this condition is true, then the for loop will run for a maximum of \( k \) time resulting in total complexity of \( O(nk^2) \) for the linear search. The computational complexity due to the two for loops in step 2 and while loop is given as \( O(nk^2) \). Hence the total computational complexity of the pseudocode is given as \( O(nk^2) \). To generate \( k \) coded packets the total complexity will be given as \( O(nk^3) \).

### D. Computational Complexity Comparison

The Hamming-D algorithm [23] runs on the packet status table, where the entry \( a_{ij} \) in the table is 1 if the \( j^{th} \) receiver has not received the \( i^{th} \) packet, and 0 otherwise. The objective of the Hamming-D algorithm is to first select two columns with the largest Hamming distance. As we intend to derive the worst case computational complexity of the Hamming-D algorithm, we consider instances of the problem, where the algorithm selects only two packets for encoding.

The number of combination of packets whose Hamming distance needs to be evaluated is given as \( \binom{k}{2} \) in the first iteration. For each pair of packets, \( n \) entries of the table need to be compared, resulting in computational complexity of \( O(nk^2) \). After the first pair of packet has been selected, encoded and transmitted, packet reception status would be collected. Hamming-D will then need to evaluate at least \( \binom{k-2}{2} \) combination of packets, with complexity of \( O(nk^2) \). Continuing this way, the total worst-case complexity of generating \( k \) coded packets is given as \( O(nk^3) \).

The computational complexities of the vertex coloring-based heuristic algorithm [8, Fig. 3] is given as \( O(nk^2) \) for constructing the graph as there are \( |V|^2 \) edges, where \( V = |V| = k \) is the set of vertices corresponding to input packets, and to construct an edge between two vertices (i.e. two lost packets) it has to be evaluated whether there exists a receiver such that it has lost one of those two packets. The complexity of sorting the vertices by their degrees is \( O(k \log k) \), and complexity due the while and for loop present in their pseudocode is \( O(k^2) \).

The vertex coloring-based heuristic algorithm is called by the heuristic-based dynamic scheme [8, Fig. 5]. For \( p > 0 \), the dynamic scheme will call the vertex coloring based heuristic algorithm given as a function of \( k \). We derive a lower bound of this function.

In the first iteration of transmission, at least one receiver would have not received an average of \( kp \) packets. To retransmit these packets, any encoding algorithm will generate at least \( kp \) coded packets. In the second iteration, an average of at least \( kp^2 \) of these coded packets would not be received by at least one receiver. Continuing this way, using the formula for summation of geometric series, it can be shown that the vertex coloring-based heuristic algorithm would be called at least \( \frac{kp}{1-p} \) times. Hence the total complexity of Vertex-Coloring Dynamic scheme is given as \( O(nk^3) \).

In the Weight-Pick algorithm [12] under the assumption of memoryless decoding, and constraint that a coded packet can contain at most \( C_{max} = 5 \) input packets, the computational complexity of a greedy search algorithm is given as \( O(nk) \). To generate \( k \) such coded packets, the total complexity is given as \( O(nk^2) \). The computational complexity of generating MDS code over the larger field size of \( GF(q = n) \) is given as \( O(n^2k^3) \) [6].

The above analysis shows that the complexity of \( O(nk^3) \) for our proposed scheme is similar to the complexity of Hamming-D scheme, and Vertex-Coloring Dynamic, and less than the computational complexity of generating MDS code. Only the computational complexity of the Weight-Pick algorithm is

**TABLE I**

**Greedy Assignment Pseudocode.**

<table>
<thead>
<tr>
<th>Input</th>
<th>Matrices ( Q_i, L_i, S_i, T_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>Assigned values of ( x_j ), and hence the coding vector ( w ).</td>
</tr>
</tbody>
</table>

% The While loop will run for a maximum of \( k \) times.

While \((\forall y_i \text{ are false) and (}\forall x_j \text{ are unassigned)})

Temporarily assign 0 and 1 to unassigned variables \( x_j \) and evaluate which assignment satisfies maximum propositional formulas \( y_i \).

**Step 1**

For \( \forall y_i \) which are unassigned

\( x_j \leftarrow 1 \)

% Initialize the number of propositional formulas which

% will be satisfied due to the temporary assignment.

Benefit\(_{y_i} = 0 \)

For \( \forall y_i \) which are unsatisfied

If element \( c_{t} \) of \( S_i \) is equal to 1

Increment\(_{Benefit\(_{y_i}\)} \)

The same above procedure is then repeated for \( x_j \leftarrow 0 \), and \( T_i \) is used in the above if condition.

Permanently assign 0 or 1 to \( x_j \) which satisfies maximum receivers (i.e. maximum Benefit\(_{y_i}\)).

**Update the matrices, as \( x_j \) has been permanently assigned a value**

**Step 2**

For \( \forall y_i \) which are unsatisfied

For \( \forall t \) rows of \( Q_i \), \% \( t \in 1, \ldots, (k - r_t) \)

The steps after the if condition represents the XOR addition of \( x_j \) to the RHS and LHS of the equation where the linear combination is equal to \( b_{t,k+1} \)

If \( b_{t,j} = 1 \)

\( b_{t,j} \leftarrow 0 \)

\% \( b_{t,j} \) is an element of \( Q_i \)

\( b_{t,k+1} = x_j \)

Decrement\(_{ct} \)

\% \( c_t \) is an element of \( L_i \)

If \( c_t = 1 \)

\% Linear search of the one unassigned variable in \( Q_i \).

For \( \forall s \) columns of \( Q_i \), except the last column

If \( b_{t,s} = 1 \) and \( b_{t,k+1} = 1 \)

\( c_t = 1 \)

\% \( c_s \) is an element of \( S_i \)

Else

If \( b_{t,s} = 1 \) and \( b_{t,k+1} = 0 \)

\( d_s = 1 \)

\( d_k = 1 \)

\% \( d_s \) is an element of \( T_i \)
one polynomial power degree less than the computational complexity of our proposed scheme. The lower encoding complexity of Weight-Pick algorithm comes at the tradeoff cost of higher energy consumption due to additional transmissions and redundant receptions.

V. RANK DISTRIBUTION TO GENERATE INNOVATIVE PACKET FOR THREE RECEIVERS

We use the results of this section to derive closed form expressions for the exact minimum number of transmissions and upper bound on the minimum number of transmissions for restricted instances of the problem in Section VI. In our analysis we restrict to the case of three receivers.

Our analysis makes use of the fact that the transmitter can generate an innovative packet if there exists a packet which is not given by the span of packets received by all unsatisfied clients. The cardinality of span($H_i$) is given as,

$$|\text{span}(H_i)| = \sum_{j=1}^{r_i} \binom{r_i}{j} = 2^{r_i} - 1. \quad (1)$$

The minus one in the equation is due to the fact that we exclude the zero coding vector from consideration. The transmitter can generate an innovative packet if the following inequality is satisfied,

$$\sum_{i=1}^{n} (2^{r_i} - 1) < |\text{span}(\mathcal{P})| = 2^k - 1. \quad (2)$$

Falseness of the above inequality does not imply that an innovative packet cannot be generated, as the above equation assumes that $|\text{span}(H_i) \cap \text{span}(H_j)| = 0, \forall i, j : i \neq j$. Clearly when $|\text{span}(H_i) \cap \text{span}(H_j)| > 0$, inequality in Equation (2) would still be correct.

There exists an innovative packet for all unsatisfied receivers, if $r_i \leq k - 2$ for two receivers, and $r_i \leq k - 1$ for another receiver, as the inequality (2) for such case can be expressed in simplified form as $2^k - 3 < |\text{span}(\mathcal{P})|$. 

Lemma 4. For $n = 3$ and rank distribution, $r_i = k - 1$ for two receivers and $r_i = k - 2$ for another receiver, there exists an innovative packet for all receivers.

Proof. Without loss of generality consider that $r_1 = r_2 = k - 1$ and $r_3 = k - 2$. We evaluate the lower bound of $|\text{span}(H_i) \cap \text{span}(H_j)|$, i.e. the minimum number of packets common in both span($H_1$) and span($H_2$)

Consider an all-zero column $\mathbf{v}$ in $M_1$, after some elementary column operations. The existence of such column follows from the fact that the rank of the columns of $M_1$ is $k - 1$. Therefore for any linear combination of row vectors in $M_1$ the entry in column $\mathbf{v}$ will always be equal to zero.

Our objective is to construct matrix $M_2$ such that $|\text{span}(H_1) \cap \text{span}(H_2)|$ is minimized, this happens when $|H_1 \cap H_2| = 0$. We therefore consider that all the vector entries in $M_2$, have an entry equal to one in column $\mathbf{v}$ for all rows.

It is easy to verify that only the addition of any even number of rows in $M_2$ will result in zero entry in column $\mathbf{v}$, and hence the resulting vector will also be present in the span of $H_1$.

Therefore the minimum number of such vectors which will be present in the span of both $H_1$ and $H_2$ will be given as,

$$\sum_{b} \binom{k-1}{b} = 2^{k-2} - 1. \quad (3)$$

Where $b$ is the set of positive even numbers such that $b \leq k - 1$.

Derivation of Equation (3) is shown in the Appendix. Then based on the inclusion-exclusion principle,

$$|\text{span}(H_1) \cup \text{span}(H_2)| \geq |\text{span}(H_1)| + |\text{span}(H_2)| - |\text{span}(H_1) \cap \text{span}(H_2)| \geq 2 \times (2^{k-1} - 1) - 2^{k-2} + 1. \quad (4)$$

The inequality sign in Equation (4) is due to the fact that we consider the lower bound of $|\text{span}(H_1) \cap \text{span}(H_2)|$ in the equation.

But so far we have not considered the span of $H_3$. Without loss of generality consider that the span of $H_3$ does not intersect with either the span of $H_1$ and $H_2$ to derive an upper bound on the cardinality of the union of the three has sets. Then based on the inclusion exclusion principle and the result of Equation 4,

$$|\text{span}(H_1) \cup \text{span}(H_2) \cup \text{span}(H_3)| \leq 2^k - 2 < |\text{span}(\mathcal{P})|.$$  

We have hence shown that for the rank distribution stated in this lemma, there exists an innovative packet for all the receivers. This completes the proof.

Theorem 2. For $n = 3$ the only rank distribution where an innovative packet for all receivers cannot necessarily be generated is given by $r_i = k - 1, \forall r_i$.

Proof. We adopt the optimization algorithm we had proposed in Section III. With $w$ appended as the last row of $M_1$, and after triangularization, column $a$ in $M_1$, column $b$ in $M_2$, and column $c$ in $M_3$ will have first $r_i$ rows equal to zero, leading to the generation of the set of propositional equations (5a)-(5c).

Where the coefficient $\alpha_a$ is equal to one if the $u$th column was added to column $h \in \{a, b, c\}$, otherwise it is equal to zero. An innovative packet for all three receivers can be generated if all the three equations are satisfied.

$$x_a \oplus \bigoplus_{i \in S} \alpha_a x_i = 1, \quad (5a)$$

$$x_b \oplus \bigoplus_{i \in S(b)} \alpha_b x_i = 1, \quad (5b)$$

$$x_c \oplus \bigoplus_{i \in S(c)} \alpha_c x_i = 1. \quad (5c)$$

It is easy to verify that the following two instances of the equations cannot be satisfied, (1) $x_a = 1, x_b = 1, x_c$ and (2) $x_a \oplus x_b = 1, x_b \oplus x_c = 1$ and $x_a \oplus x_c = 1$. Hence there exist instances of equations (5a)-(5c) where an innovative packet for all receivers cannot be generated when $r_i = k - 1, \forall r_i$. This completes the proof.

Corollary 1. For $n = 3$, and rank distribution $r_i = k - 1, \forall r_i$, there exists a coded packet such that the coded packet is innovative for at least two receivers.
Proof. Based on the results of Theorem 2 we know that for \(n = 3, r_i = k - 1\), \(\forall r_j\), the transmitter cannot necessarily generate an innovative packet for all three receivers, but based on the results of [6, Theorem 3] it is known that a coded packet can be constructed in polynomial time which will be innovative for at least \(q\) receivers. As we consider field size of \(q = 2\), for \(n = 3\) there exists a coded packet which will be innovative for at least two receivers. Hence at most one receiver will need to receive redundant packet(s) for rank distribution \(r_i = k - 1\), \(\forall r_j\). This completes the proof. \(\square\)

VI. MARKOV CHAIN MODEL

In this section we use Markov chain model to derive closed form expressions for restricted classes of the GIP\(_2\) problem for the ordered pair \((n, k)\) given as \((3, 2), (3, 3)\) and \((4, 2)\). This way our results demonstrate the effect of increasing packet size \(k\) and number of receivers \(n\) on retransmission rate.

In all our proposed Markov chain models, we have an absorbing state, which represents the state where all receivers are satisfied. Finding the expected number of steps to reach the absorbing state in the Markov chain can be found using method given in standard textbook such as [3, Section 7.4] and is equivalent to finding the expected number of transmissions \(E[t]\) before all receivers are satisfied using an optimal erasure code over \(GF(2)\).

The number of unique packets which are linearly dependent for at least one receiver is denoted by \(T_d\), \(T_d < \text{span} (\mathcal{P})\). The term \(T_d\) is used to distinguish between states having the same rank distribution but different set of unique packets in has sets. Element \(a_{ij}\) in the transition matrix represents the transition probability from state \(S_i\) to \(S_j\).

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_0)</td>
<td>(r_i = 0) for all receivers.</td>
</tr>
<tr>
<td>(S_1)</td>
<td>(r_i = 1) for one client, and (r_k = 0) for two receivers.</td>
</tr>
<tr>
<td>(S_2)</td>
<td>(r_i = 1) for two receivers, and (r_k = 0) for one client, and (T_d = 1).</td>
</tr>
<tr>
<td>(S_3)</td>
<td>(r_i = 2) for two receivers, and (r_k = 0) for one client.</td>
</tr>
<tr>
<td>(S_4)</td>
<td>(r_i = 1) for two receivers, (r_k = 1) for one client, and (T_d = 2).</td>
</tr>
<tr>
<td>(S_5)</td>
<td>(r_i = 1) for all receivers, and (T_d \leq 2).</td>
</tr>
<tr>
<td>(S_6)</td>
<td>(r_i = 2, r_k = 1), and (r_k = 0) for each of the receivers.</td>
</tr>
<tr>
<td>(S_7)</td>
<td>(r_i = 1) for all receivers, and (T_d = 3).</td>
</tr>
<tr>
<td>(S_8)</td>
<td>(r_i = 2) for one client, and (r_k = 1) for two receivers.</td>
</tr>
<tr>
<td>(S_9)</td>
<td>(r_i = 2) for two receivers, and (r_k = 0) for one client.</td>
</tr>
<tr>
<td>(S_{10})</td>
<td>(r_i = 2) for two receivers, and (r_k = 0) for one client.</td>
</tr>
<tr>
<td>(S_{11})</td>
<td>(r_i = 2) for all receivers. Absorbing state.</td>
</tr>
</tbody>
</table>

**TABLE II**

Markov states description for \(n = 3, k = 2\).

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_0)</td>
<td>(r_i = 0) for all receivers.</td>
</tr>
<tr>
<td>(S_1)</td>
<td>(r_i = 0) for two receivers, and (r_k = 1) for one client.</td>
</tr>
<tr>
<td>(S_2)</td>
<td>(r_i = 1) for two receivers, (r_k = 0) for one client, and (T_d = 1).</td>
</tr>
<tr>
<td>(S_3)</td>
<td>(r_i = 2) for two receivers, and (r_k = 0) for two client.</td>
</tr>
<tr>
<td>(S_4)</td>
<td>(r_i = 1) for two receivers, (r_k = 1) for one client, and (T_d = 2).</td>
</tr>
<tr>
<td>(S_5)</td>
<td>(r_i = 2, r_k = 1), and (r_k = 0) for each of the receivers and (T_d = 3).</td>
</tr>
<tr>
<td>(S_6)</td>
<td>(r_i = 2, r_k = 1), and (r_k = 0) for each of the receivers and (T_d = 4).</td>
</tr>
<tr>
<td>(S_7)</td>
<td>(r_i = 0) for two receivers, and (r_k = 3) for one client.</td>
</tr>
<tr>
<td>(S_8)</td>
<td>(r_i = 1) for all receivers, and (T_d = 1).</td>
</tr>
<tr>
<td>(S_9)</td>
<td>(r_i = 1) for all receivers, and (T_d = 2).</td>
</tr>
<tr>
<td>(S_{10})</td>
<td>(r_i = 1) for all receivers and (T_d = 3).</td>
</tr>
<tr>
<td>(S_{11})</td>
<td>(r_i = 1) for two receivers, and (r_k = 2) for one client.</td>
</tr>
<tr>
<td>(S_{12})</td>
<td>(r_i = 2) for two receivers, (r_k = 0) for one client, and (T_d \neq 5).</td>
</tr>
<tr>
<td>(S_{13})</td>
<td>(r_i = 3, r_k = 1), and (r_k = 0) for each of the receivers.</td>
</tr>
<tr>
<td>(S_{14})</td>
<td>(r_i = 2) for two receivers, (r_k = 0) for one client and (T_d = 5).</td>
</tr>
<tr>
<td>(S_{15})</td>
<td>(r_i = 1) for two receivers, (r_k = 2) for one client and (T_d = 4).</td>
</tr>
<tr>
<td>(S_{16})</td>
<td>(r_i = 1) for two receivers, (r_k = 2) for one client and (T_d = 3) or (T_d = 5).</td>
</tr>
<tr>
<td>(S_{17})</td>
<td>(r_i = 2) for two receivers, and (r_k = 1) for one client.</td>
</tr>
<tr>
<td>(S_{18})</td>
<td>(r_i = 1) for two receivers, and (r_k = 3) for one client.</td>
</tr>
<tr>
<td>(S_{19})</td>
<td>(r_i = 3, r_k = 2), and (r_k = 0) for each of the receivers.</td>
</tr>
<tr>
<td>(S_{20})</td>
<td>(r_i = 2) for two receivers, (r_k = 1) for one client and (T_d = 5) or (T_d = 6).</td>
</tr>
<tr>
<td>(S_{21})</td>
<td>(r_i = 2) for all receivers, and (2 \leq T_d \leq 6).</td>
</tr>
<tr>
<td>(S_{22})</td>
<td>(r_i = 3, r_k = 2), and (r_k = 1) for each of the receivers.</td>
</tr>
<tr>
<td>(S_{23})</td>
<td>(r_i = 3) for two receivers, and (r_k = 0) for one client.</td>
</tr>
<tr>
<td>(S_{24})</td>
<td>(r_i = 2) for all receivers and (T_d = 7).</td>
</tr>
<tr>
<td>(S_{25})</td>
<td>(r_i = 2) for two receivers, and (r_k = 3) for one client.</td>
</tr>
<tr>
<td>(S_{26})</td>
<td>(r_i = 3) for two receivers, and (r_k = 1) for one client.</td>
</tr>
<tr>
<td>(S_{27})</td>
<td>(r_i = 3) for two receivers, and (r_k = 2) for one client.</td>
</tr>
<tr>
<td>(S_{28})</td>
<td>(r_i = 3) for all receivers. Absorbing state.</td>
</tr>
</tbody>
</table>

**TABLE III**

Transition matrix for \(n = 3, k = 2\).

**TABLE IV**

Markov states description for \(n = 3, k = 3\).

**A. Markov chain model for \(k = 2, n = 3\)**

For \(k = 2\), \(\text{span}(\mathcal{P}) = \{P_1, P_2, P_1 \oplus P_2\}\). A transmitter cannot transmit an innovative packet for all the three receivers iff one of the receivers has \(P_1\), another has \(P_2\), and the third receiver has \(P_1 \oplus P_2\), which is represented by state \(S_7\) in our Markov chain model. We distinguish \(S_7\) from \(S_3\) where a transmitter may be able to generate an innovative packet, e.g. when all three receivers have \(P_1\). A self-explanatory description of the other states is given in Table II, and a transition matrix is given in Table III.

**B. Markov chain for \(k = 3, n = 3\)**

A transmitter cannot transmit an innovative packet for three receivers iff \(r_i = 2\) for all receivers and \(T_d = 7\), represented by state \(S_{24}\). A description of the other states is given in Table IV, and the transition matrix is shown in Table V.

We briefly explain some of these states. State \(S_{15}\) for example could represent a scenario when \(R_1\) and \(R_2\) has \(P_1\), and \(R_3\) has \(P_2\) and \(P_3\) resulting in \(T_d = 4\). While \(S_{10}\) could represent a scenario when \(R_1\) and \(R_2\) has \(P_1\), and \(R_3\) has \(P_1\) and \(P_2\) resulting in \(T_d = 3\).

**C. Markov chain for \(k = 2, n = 4\)**

In this scenario there are three rank distributions for which a transmitter cannot transmit an innovative packet and these distributions are represented as states \(S_8, S_{11}\) and \(S_{14}\) in the...
Markov chain model. A description of the Markov chain’s states is given in Table VI, and the transition matrix is shown in Table VII.

D. Throughput Performance

The results of Markov chain is compared with the retransmission rate $R_{0}$ of MDS code. For simulation carried to verify the correctness of our Markov chain analytical result, the transmitter chooses a packet which is innovative for maximum number of unsatisfied receivers.

Figure 2 and 3 show the effect of increasing the number of packets and receivers respectively on $R_{0}$. The simulation results matches with the Markov chain results and verify the correctness of the Markov chain analytical model. Based on the result of our numeric analysis and greedy scheme on the optimization algorithm we present the following conjectures.

**Conjecture 1.** Based on the results of Figure 2 as $k \to \infty$, there exists an erasure code over $GF(2)$ whose throughput performance approaches the throughput performance of MDS code.

**Conjecture 2.** Based on the results of Figures 1 and 3, for a fixed $n$ the GIIP problem can be approximated within a constant multiplicative factor of the optimal value.

E. Upper Bound for Three Receivers

In this section we derive an upper bound on the minimum number of transmissions needed to satisfy all three receivers. We first derive the expected value of redundant packets $\delta$ one of the receiver need to receive. As we intend to derive an upper bound, in our derivation we assume that the network enters a state where $r_{i} = k - 1$, if $r_{i}$, such that the transmitter cannot transmit an innovative packet for all receivers. In such case as shown in Corollary 1 there exists an erasure code over $GF(2)$, such that at most one receiver receive redundant packets, while the other two receivers receive innovative packet. Then the probability $P[\delta = \beta]$ that one of the receiver receives $\beta$ redundant packets is given by,

$$P[\delta = \beta] = (sp^{2})^{\beta-1}(sp^{2} + 2s^{2}p + s^{3})^\beta,$$

and the expected value of $\delta$, $E[\delta]$, is given as,

$$E[\delta] = \sum_{\beta=1}^{\infty} \beta \cdot P[\delta = \beta]$$

$$= (sp^{2} + 2s^{2}p + s^{3}) \cdot \sum_{\beta=1}^{\infty} \beta(sp^{2})^{\beta-1}$$

$$= sp^{2} + 2s^{2}p + s^{3} \cdot \frac{(1 - sp^{2})^{2}}{1 - sp^{2}}.$$ 

Where $0 < E[\delta] \leq 1$ for $0 \leq p < 1$. We now quantify the expected number of transmissions such that two receivers receive $k$ packets and another receiver receive $k+1$ packets.
Let $\ell$ denote the number of transmissions before all receivers receive at least $k$ innovative packets. Denote by $D(m)$ the probability that $\ell \leq m$, i.e. $P[\ell \leq m]$. The probability that a receiver receives $j$ packets out of $m$ transmissions follows Bernoulli distribution and is given as, 

$P[X = j]$.

The probability that the receiver receives at least $k$ innovative packets after $m$ transmissions is given as,

$$D_1(m) = \sum_{j=k}^{m} \binom{m}{j} s^j p^{m-j},$$  \hspace{1cm} (8)

and the probability that the receiver receives at least $k$ innovative packets and one dependent packet (i.e. $k + 1$ packets) after $m$ transmissions is given as,

$$D_2(m) = \sum_{j=k+1}^{m} \binom{m}{j} s^j p^{m-j}. \hspace{1cm} (9)$$

As the packet reception at each of the receivers is independent, the probability $D(m)$ is given as, $D_1(m) \cdot D_2(m)$. The expected number of transmissions to transmit $k$ packets to two receivers and $k + 1$ packets to another receiver is given as,

$$E[\ell] = \sum_{m=0}^{\infty} \left(1 - D(m)\right) = k + 1 + \sum_{m=k+1}^{\infty} \left(1 - D(m)\right). \hspace{1cm} (10)$$

The results so far assume that the network will always enter in a state where $r_i = k - 1, \forall r_i$. This may not necessarily be true. We next evaluate the probability $U_d$ that a network enters in such a state for the first time. Clearly $U_d$ will decrease as $k$ increases, as the number of different rank distribution increases with $k$. Therefore $U_d$ for $k = 2$ serves as an upper bound of $U_d$ for arbitrary $k$.

We henceforth adopt the Markov chain model introduced in Section VI-A. Then the upper bound for the minimum number of expected transmissions for three receivers will be given as,

$$E_m[\ell] = (1 - U_d) \times E_m[\ell] + U_d \times E[\ell]. \hspace{1cm} (11)$$

Where $E_m[\ell]$ is the expected number of transmission for MDS code. The derivation of $E_m[\ell]$ can be found in [10]. The correctness of Equation (11) follows from Theorem 2, i.e. if the network does not enter a state given as $r_i = k - 1, \forall r_i$, then an innovative packet for all receivers can be generated.

We define “transmission failure” to be a scenario where the network enters the same state (known as self-loop), and denote the number of times the Markov chain enters self-loop by $t$. Assuming no transmission failure takes place, then the probability $\sigma$ of entering in dull state $S_7$ (see Section VI-A) is given as,

$$\sigma = a_{01} \times a_{14} \times a_{47} = 3sp^2 \times 2sp^2 \times sp^2 = 6s^3p^{10}. \hspace{1cm} (12)$$

However to calculate $U_d$ we need to consider self-loop due to transmission failure. Assuming one transmission failure, then the probability of entering the dull state is given by $3\sigma p^3$, i.e., the summation of probabilities that the self-loop occurs at any one of the states $S_0$, $S_1$, or $S_4$. For arbitrary $t$, the number of selections to assign $t$ transmission failures at $S_0$, $S_1$ and $S_4$ is given by the combination $\binom{t - 1}{2}$. Based on this approach, the value of $U_d$ is given as,

$$U_d = \sum_{t=0}^{\infty} \binom{t + 2}{2} \sigma p^{3t}. \hspace{1cm} (13)$$

A plot of the upper bound of the exact minimum number of transmissions achievable using an optimal $GF(2)$ erasure code is plotted in Figure 4 using Equation (11). The result is compared with the retransmission rate of MDS code (lower bound), and RL code over $GF(2)$ using simulation as the analytical model of RL code is unknown [17]. The results of the graph shows that for $n = 3$ there exists an erasure code over $GF(2)$ whose throughput performance is virtually similar to the throughput performance of MDS code.

VII. CONCLUSION

In this paper we studied the characteristics of optimal erasure codes over $GF(2)$ for single-hop wireless multicasting. We first showed that constructing such an optimal code is an NP-Complete problem by showing a mapping scheme which maps the optimization problem to an NP-Complete Boolean constraint satisfaction problem (CSP). We proposed efficient greedy encoding scheme for the problem, which was shown...
to be significantly better than previous state of the art coding schemes and performed very close to the optimal MDS code. To the best of our knowledge this is the first work of its kind where we study non-asymptotic performance of such optimal erasure codes.

The analytical results for restricted instances shed interesting insight on the effect of increasing number of packets and network size on the throughput performance of such an optimal erasure code, and how the throughput of such an optimal code compares with the optimal MDS code. The analytical method which we have presented to derive the upper bound for three receivers also serves as a proof-of-concept to derive upper bound for arbitrary number of receivers at the cost of increasing mathematical complexity.

REFERENCES


In this Appendix we demonstrate the derivation of Equation (3). We make use of the following relation,

$$\left( \frac{f}{a} \right) = \left( \frac{f-1}{a-1} \right) + \left( \frac{f-1}{a} \right).$$

Then Equation (3) can be rewritten as,

$$\sum_b \binom{k-1}{b} = \binom{k-1}{2} + \binom{k-1}{4} + \binom{k-1}{6} + \cdots$$

$$= \binom{k-2}{1} + \binom{k-2}{2} + \binom{k-2}{3} + \cdots + \binom{k-2}{k-3} + \binom{k-2}{k-2}$$

$$= 2^{k-2} - 1$$