This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

Minimum Broadcast Decoding Delay for Generalized Instantly Decodable Network Coding

Sameh Sorour and Shahrokh Valaee
The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto
Toronto, ON, M5S 3G4, Canada
Email:{samehsorour, valaee}@comm.utoronto.ca

***Abstract*—In this paper, we introduce the concept of generalized instantly decodable network coding (G-IDNC) to further minimize decoding delay in wireless broadcast, compared to strict instantly decodable network coding (S-IDNC), studied in [1], [2]. G-IDNC loosens the strict instant decodability constraint in order to target more receivers while preserving the attractive properties of S-IDNC. We show that the minimum decoding delay problem for G-IDNC can be formulated as a maximum weight clique problem over a well structured graph. Since finding the maximum weight clique of a graph is NP-hard, we design a simple heuristic G-IDNC algorithm with sub-optimal performance. However, simulation results show that both proposed optimal and heuristic G-IDNC algorithms considerably outperform several other S-IDNC and G-IDNC optimal and heuristic approaches.**

***Index Terms*—Wireless Broadcast; Network Coding; Minimum Decoding Delay; Maximum Weight Clique Problem.**

I. Introduction

Network coding (NC) has shown great abilities to improve the performance of wireless broadcast applications, in which receivers require quick and reliable reception of packets through erasure channels. Different NC approaches were proposed in the literature based on the type of application. In this paper, we are interested in a category of applications, in which decoded packets should bring new information to the receivers, irrespective of their order. Examples of such applications are real-time streams encoded using multiple description source coding, roadside to vehicle safety message broadcast and coordinated command dissemination to sensors. In such applications, every packet detected by a receiver, without bringing new information, increases its decoding delay.

The design of online NC algorithms, which minimize decoding delay over broadcast erasure channels, has been an attractive area of research [1]–[5] in the past two years. In [3], the authors proposed an NC algorithm for the three-receiver case, proved its rate optimality and conjectured that it achieves an asymptotically optimal average delay. In [4], [5], the decoding delay performance of several online NC algorithms were compared for i.i.d. erasure channels. However, the proposed algorithms performed un-prioritized packet selection for each NC transmission and did not consider the channel conditions in their selection procedures.

In [1], [2], the minimum decoding delay problem was studied for a subclass of NC called *strict instantly decodable network coding (S-IDNC)*. This class of NC forces all coded packets to include at most one missing source packet for each receiver. The interest in S-IDNC arises from its numerous desirable properties [1]. First, its instant decodability property allows the instant use of the packets upon their reception at the receivers, which is the main requirement in the applications of interest. Moreover, S-IDNC encoding can be implemented using binary XOR, which eliminates the complicated operations over large Galois fields and the coefficient reporting overhead. This XOR encoding also simplifies the decoding process at the receivers, as each receiver can simply cancel out the packets it already knows. This eliminates the need for matrix inversion at the receivers, which is a computational bottleneck in linear and random NC [1]. Finally, no buffers are needed at the receivers to store coded packets for future decoding possibilities. These simple decoding and no-buffer properties allow the design of simple and cost-efficient receivers.

For this class of NC, [1], [2] formulated the minimum decoding delay problem as a set packing problem and proposed optimal and heuristic NC algorithms to solve it. The proposed algorithms were shown to outperform the equivalent random selection algorithm in [4], [5]. However, the strict instant decodability condition of S-IDNC in [1], [2] limits the coding opportunities in each transmission, as it forbids packet combinations violating it even for only one receiver. Obviously, this restriction in coding opportunities can reduce the limits of decoding delay minimization.

In this paper, we address the following questions: *Is there an NC scheme that can achieve a lower decoding delay while preserving all the benefits of S-IDNC? And if this scheme exists, how can we formulate and solve the minimum decoding delay problem for it?* To answer the former question, we first introduce the concept of *Generalized Instantly Decodable Network Coding (G-IDNC)*, in which the strict instant decodability constraint in [1] is loosened at the sender in order to target more receivers per transmission, while keeping the instant decodability constraint at the receivers. Consequently, G-IDNC is expected to achieve lower decoding delay compared to S-IDNC, while keeping all its attractive properties. We then answer the second question by showing that the minimum decoding delay problem in G-IDNC can be formulated as a maximum weight clique problem over a well structured graph. Since finding the maximum weight clique in a graph is NP-hard, we design a simple maximum weight vertex search algorithm, with sub-optimal performance, but much lower complexity. Finally, we employ extensive simulations to quantify the reduction in decoding delay achieved by our proposed optimal and heuristic G-IDNC algorithms compared to several other optimal and heuristic S-IDNC and G-IDNC approaches.

The rest of the paper is organized as follows. In Section II, we introduce the system model and parameters. The S-IDNC formulation proposed in [1], [2] is briefly described in Section III. In Section IV, we describe our proposed G-IDNC scheme and introduce its minimum decoding delay formulation. Our proposed G-IDNC heuristic algorithm is then described in Section V. Simulation results are discussed in Section VI. Finally, Section VII concludes the paper.

II. System Model and Parameters

The model consists of a wireless sender that is required to deliver a frame (denoted by $\mathcal{N}$) of N source packets to a set (denoted by $\mathcal{M}$) of M receivers. The sender initially transmits the N packets of the frame uncoded in an *initial phase*. Each receiver feedbacks to the sender a positive/negative acknowledgement (ACK/NAK) for each received/lost packet. At the end of the initial phase, two sets of packets are attributed to each receiver i:

- The *Has* set (denoted by $\mathcal{H}_i$) is defined as the set of packets correctly received by receiver i in the initial phase of the current broadcast frame.
- The *Wants* set (denoted by $\mathcal{W}_i$) is defined as the set of packets that are lost by receiver i in the initial phase of the current broadcast frame. In other words, $\mathcal{W}_i = \mathcal{N} \setminus \mathcal{H}_i$.

The sender stores this information in a *feedback matrix* $\mathbf{F} = [f_{ij}]\,,\ \forall\ i \in \mathcal{M}, j \in \mathcal{N}$ such that:

$$f_{ij} = \begin{cases} 0 & j \in \mathcal{H}_i \\ 1 & j \in \mathcal{W}_i\ . \end{cases} \tag{1}$$

After the initial phase, a packet recovery phase starts. In this phase, the sender exploits the diversity of received and lost packets at different receivers to transmit network coded combinations of the source packets. For each transmission, the receivers send ACK/NAK for each decoded/undecoded packet, which are used by the sender to update the feedback matrix and the Has/Wants sets of all receivers. This process is repeated until all receivers obtain all the packets. Let p_i, $i \in \mathcal{M}$, be the packet erasure probability of receiver i, assumed to be constant during a frame delivery period.

In the recovery phase, the transmitted coded packets can be one of the following three options for each receiver i:

- *Non-Innovative*: A packet is non-innovative for receiver i if it contains *no source packets* from $\mathcal{W}_i$.
- *Instantly Decodable*: A packet is instantly decodable for receiver i if it contains *only one source packet* from $\mathcal{W}_i$.
- *Non-Instantly Decodable*: A packet is non-instantly decodable for receiver i if it contains two or more source packets from $\mathcal{W}_i$.

We define the decoding delay similar to [1], [2] as follows:

Definition 1. *At any recovery phase transmission, a receiver i, with non-empty Wants set, experiences a one unit increase of decoding delay if it successfully receivers a packet that is either non-innovative or non-instantly decodable.*

In the rest of the paper, we will term the receivers, for which a packet is instantly decodable, as targeted receivers.

III. Strict Instantly Decodable Network Coding (S-IDNC)

A. Description and Formulation

As the name declares, the strict instantly decodable network coding (S-IDNC) scheme forces the sender to transmit packets that are only either non-innovative or instantly decodable for all receivers, in each recovery phase transmission. This S-IDNC property provides important benefits as explained in Section I.

Sadeghi *et al.* [1] formulated the S-IDNC minimum decoding delay problem as follows. Let $\mathbf{w_p} = \left[w_j^p\right], j \in \mathcal{N}$, be the packet weight vector, such that w_j^p is equal to the number of receivers having packet j in their Wants sets. Consequently, the minimum decoding delay problem for S-IDNC can be expressed in the form of a set packing problem as follows:

$$\max_{\mathbf{x}}\ \mathbf{w_p}^T\mathbf{x} \tag{2a}$$

$$\text{subject to}\quad \mathbf{F}\mathbf{x} \leq \mathbf{1}_M \tag{2b}$$

$$\mathbf{x} \in \{0,1\}^N\ , \tag{2c}$$

where $\mathbf{1}_M$ is the all-ones vector of length M. An important remark about this formulation is that its objective function (2a) represents the total number of targeted receivers, and thus its maximization is expected to reduce the decoding delay.

B. Limitations

Consider the following example of feedback matrix, where the rows and columns represent receivers and packets, respectively:

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\ . \tag{3}$$

It can be easily seen that packets 1, 2 and 3$\oplus$4 are the only ones satisfying Constraint (2b). Since each of these packets targets three receivers, each of the other two receivers will experience a decoding delay increase, if it successfully receives the packet. On the other hand, the coded packet 1$\oplus$2, violating the S-IDNC constraint, targets four receivers, and thus only one receiver can suffer from a decoding delay increase. Consequently, this packet is expected to achieve lower sum decoding delay than those constrained by S-IDNC. However, it belongs to another class of NC, which may impose different coding, decoding and receiver requirements. In the next section, we will describe a generalized IDNC scheme that can use this packet while preserving all benefits of S-IDNC.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

IV. Generalized Instantly Decodable Network Coding (G-IDNC)

A. Description of G-IDNC

As shown in the previous section, the instant decodability constraint in (2b) is limiting the coding opportunities and thus limiting the reduction of decoding delay. One way to solve this limitation is to loosen this constraint at the sender algorithm, in order to increase the number of targeted receivers in each transmission and thus potentially reduce the decoding delay.

An example of such case is the transmission of packet $1 \oplus 2$ in example (3), which is non-instantly decodable for receiver 3. The straightforward option for receiver 3 is to keep the packet for future possible use, if it receives packets 1,2, or even another combination of packets 1 and 2 if operations on higher Galois fields are allowed. This option will potentially decrease the decoding delay of this receiver if it receives a future linear combination of packets 1 and 2. However, it will require decoding buffers at the receivers and more complicated coding and decoding processes, which eliminates all the benefits of S-IDNC.

Since we want to keep these S-IDNC benefits, we propose another option in which the receivers discard all detected packets that are non-instantly decodable. In this way, the encoding and decoding processes can still be done in binary XOR and the receivers will not need decoding buffers. Although this proposed scheme loosens the instant decodability constraint at the sender, all receiver decoding is still from instantly received packets. Consequently, we will refer to this proposed scheme as the *Generalized Instantly Decodable Network Coding (G-IDNC)*. It can be easily inferred that G-IDNC allows more efficient packet combinations at the sender, while preserving all the benefits of S-IDNC. The question now is how to generate such G-IDNC packets so as to minimize the mean decoding delay. This will be the focus of the next sections.

B. G-IDNC Packet Generation

To minimize decoding delay in G-IDNC, we should first explore all possible packet combinations that are instantly decodable by any subset or all the receivers. In general, two packets in the Wants sets of two receivers are combinable in G-IDNC, if these two receivers can instantly decode the resulting coded packet. This can only happen if these two receivers are requesting the same packet or if the packet wanted by each receiver is in the Has set of the other. Now, we have to find a representation of all such combinations between all packets in the Wants sets of all receivers. This can be represented in the form of a graph model, first introduced in the context of a heuristic algorithm solving the index coding problem [6], [7]. In our context, we will call this graph as the *G-IDNC graph*.

The G-IDNC graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is constructed by first generating a vertex $v_{ij} \in \mathcal{V}$ for each packet $j \in \mathcal{W}_i$, $\forall\ i \in \mathcal{M}$. Two vertices v_{ij} and v_{kl} in $\mathcal{V}$ are connected by an edge in $\mathcal{E}$ if one of the following conditions is true:

- C1: $j = l \Rightarrow$ The two vertices are induced by the request of the same packet j by two different receivers i and k.
- C2: $j \in \mathcal{H}_k$ and $l \in \mathcal{H}_i \Rightarrow$ The requested packet of each vertex is in the Has set of the receiver that induced the other vertex.

From the above connectivity conditions, it is clear that an edge generated by C2, between two vertices v_{ij} and v_{kl}, represents a possible combination of packets j and l that will be instantly decodable for receivers i and k. C1-generated edges do not involve packet combinations but express the interest of the two receivers in the same packet.

Given this graph formulation, we can easily define the set of all feasible packet combinations in G-IDNC as the set of packet combinations defined by all maximal cliques in $\mathcal{G}$ (a maximal clique is a clique that is not a subset of any larger cliques). Consequently, the sender can generate a G-IDNC packet for a given transmission by XORing all the packets identified by the vertices of a selected maximal clique in $\mathcal{G}$. In the next section, we will determine the maximal clique selection policy that the sender should follow to minimize the expected increase in the sum decoding delay of all receivers.

C. Minimum Decoding Delay Formulation for G-IDNC

Let κ be a maximal clique in the G-IDNC graph chosen for a given transmission, and let $d_i(\kappa)$ be the decoding delay increase experienced by receiver i for this transmission. If receiver i has a vertex in κ, it will never experience a decoding delay increase (i.e. $d_i(\kappa) = 0$). If not, then $d_i(\kappa)$ can be either 0 or 1 with the following probabilities:

$$\mathbb{P}\left[d_i(\kappa) = 0\right] = p_i \tag{4}$$

$$\mathbb{P}\left[d_i(\kappa) = 1\right] = 1 - p_i\ . \tag{5}$$

Let $D(\kappa)$ be the sum of the decoding delay increases of all receivers after this transmission $\left(\text{i.e. } D(\kappa) = \sum_{i=1}^{M} d_i(\kappa)\right)$. Define $\mathcal{T}(\kappa)$ as the set of targeted receivers in this transmission (i.e. the set of receivers having vertices in κ), and $\mathcal{M}_w$ as the set of receivers having non-empty Wants sets.

According to these definitions, the expected sum decoding delay increase after this transmission can be expressed as:

$$\mathbb{E}\left[D(\kappa)\right] = \sum_{i \in \mathcal{M}_w \setminus \mathcal{T}(\kappa)} (1 - p_i)\ . \tag{6}$$

Consequently, we can formulate the minimum mean decoding delay problem in G-IDNC as a clique selection algorithm, such that:

$$\begin{aligned} \kappa^* &= \arg\min_{\kappa \in \mathcal{G}} \left\{\mathbb{E}\left[D(\kappa)\right]\right\} \\ &= \arg\min_{\kappa \in \mathcal{G}} \left\{ \sum_{i \in \mathcal{M}_w \setminus \mathcal{T}(\kappa)} (1 - p_i) \right\} \\ &= \arg\max_{\kappa \in \mathcal{G}} \left\{ \sum_{i \in \mathcal{T}(\kappa)} (1 - p_i) \right\}\ . \end{aligned} \tag{7}$$

In other words, the minimum decoding delay problem in G-IDNC is equivalent to a maximum weight clique problem on the G-IDNC graph, in which the weight of each vertex is the packet reception success probability of its inducing receiver.

It is well known that the maximum weight clique problem is NP-hard [8], and is hard to approximate [9]. On the other hand, there exist several polynomial time algorithms solving this problem for moderate size graphs ([10] and references therein). However, the complexity of these algorithms is still prohibitive for the applications of interest in this paper [10]. Consequently, we will design a simple heuristic in Section V to solve the problem with much lower complexity.

V. Proposed heuristic G-IDNC Algorithm

In this section, we design a simple algorithm that performs clique selection, in linear time with the G-IDNC graph size (i.e. $O(|\mathcal{V}|)$), using a maximum weight vertex search. For this search to be efficient, the vertices' weights must not only reflect the reception success probabilities of their inducing receivers, but also their connection to vertices having high reception success probabilities.

To design these vertices' weights, we first define $a_{ij,kl}$ as the adjacency indicator of vertices v_{ij} and v_{kl} in $\mathcal{G}$ such that:

$$a_{ij,kl} = \begin{cases} 1 & v_{ij} \text{ is connected to } v_{kl} \text{ in } \mathcal{G} \\ 0 & \text{otherwise .} \end{cases} \tag{8}$$

We then define the weighted degree Δ_{ij} of vertex v_{ij} as:

$$\Delta_{ij} = \sum_{\forall v_{kl} \in \mathcal{G}} a_{ij,kl}(1 - p_k) \ . \tag{9}$$

Thus, a large weighted vertex degree reflects its connection to a large number of vertices belonging to receivers with large reception success probabilities (or low erasure probabilities). We finally define the vertex weight w_{ij} as:

$$w_{ij} = \Delta_{ij}(1 - p_i) \ . \tag{10}$$

Consequently, a vertex v_{ij} has a large weight when it both belongs to a receiver with low erasure probability, and is connected to a large number of vertices having low erasure probabilities. Let $\mathcal{G}_v(\mathcal{V}_v, \mathcal{E}_v)$ be the subgraph in $\mathcal{G}$ only including all vertices connected to vertex v and the edges connecting these vertices to each other.

Based on these definitions, we can introduce our proposed packet selection algorithm for G-IDNC as depicted in Algorithm 1. At first, κ^* is an empty set. The algorithm starts by selecting the maximum weight vertex in $\mathcal{G}$ to be the source vertex in κ^*. For each of the following iterations, the algorithm first recomputes the new vertex weights, within the subgraph connected to all previously selected vertices in κ^*, then adds the new maximum weight vertex to it. The algorithm stops when there is no further vertex connected to all vertices in κ^*. We refer to this algorithm as the *maximum weight vertex search algorithm*. Once the clique is computed, the sender forms and sends the coded packet, by XORing the source packets identified by the vertices in κ^*.

VI. Simulation Results

In this section, we present simulation results comparing the performance of our proposed optimal and heuristic G-IDNC algorithms (denoted by G-Copt and G-Csrh, respectively, in the figure legends) to:

- Random clique search algorithms for both S-IDNC and G-IDNC (denoted by S-rnd and G-rnd, respectively), which are comparable to the algorithms tested in [4].
- Optimal packet weighted selection algorithms for both S-IDNC (proposed in [2]) and G-IDNC (denoted by S-Popt and G-Popt, respectively).
- Heuristic packet weighted selection algorithms for both S-IDNC (proposed in [2]) and G-IDNC (denoted by S-Psrh and G-Psrh, respectively).

We assume that the mean decoding delays of different receivers are computed per frame, and then these mean delays are averaged over a large number of iterations for each point. From frame to frame, we assume that the packet erasure probabilities of different receivers change uniformly in a given range, while keeping the average erasure probability (p) constant.

Figure 1 depicts the comparison of mean decoding delays achieved by the different algorithms against M, for $N = 30$ and $p = 0.15$. The figure shows that our proposed G-Copt algorithm outperforms the S-Popt and G-Popt algorithms by $57-69\%$ and $45-51\%$, respectively. It also shows that our proposed G-Csrh algorithm outperforms the optimal S-Popt and G-Popt algorithms by $15-59\%$ and $2-26\%$, respectively.

Algorithm 1 Maximum Weight Vertex Search Algorithm

Require: **F** and $p_i \ \forall \ i \in \mathcal{M}$
  Initialize $\kappa^* = \varnothing$.
  Construct $\mathcal{G}(\mathcal{V}, \mathcal{E})$.
  while $\mathcal{V} \neq \varnothing$ **do**
    Compute Δ_{ij} and w_{ij} using (9) and (10).
    Select $v^* = \arg\max_{v_{ij} \in \mathcal{G}} \{w_{ij}\}$.
    Set $\kappa^* \leftarrow \kappa^* \cup v^*$.
    Set $\mathcal{G}(\mathcal{V}, \mathcal{E}) \leftarrow \mathcal{G}_{v^*}(\mathcal{V}_{v^*}, \mathcal{E}_{v^*})$.
  end while

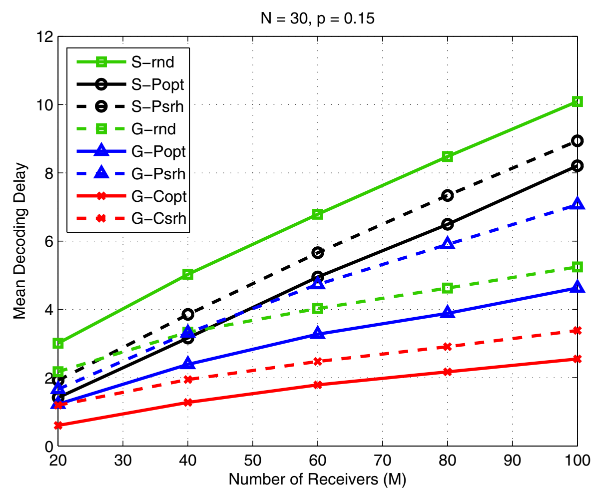

Fig. 1. Comparison of Mean Decoding Delays vs M.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

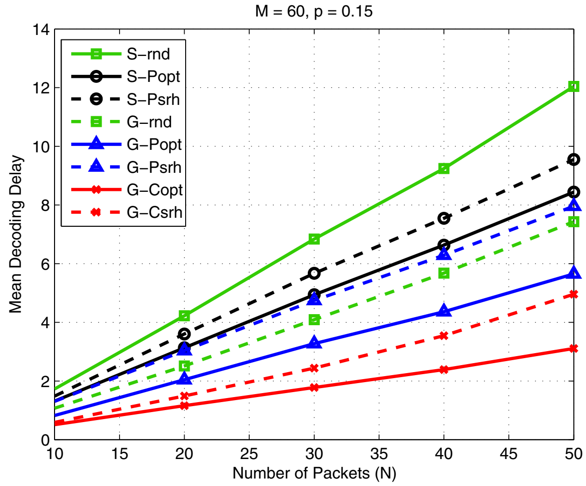

Fig. 2. Comparison of Mean Decoding Delays vs N.

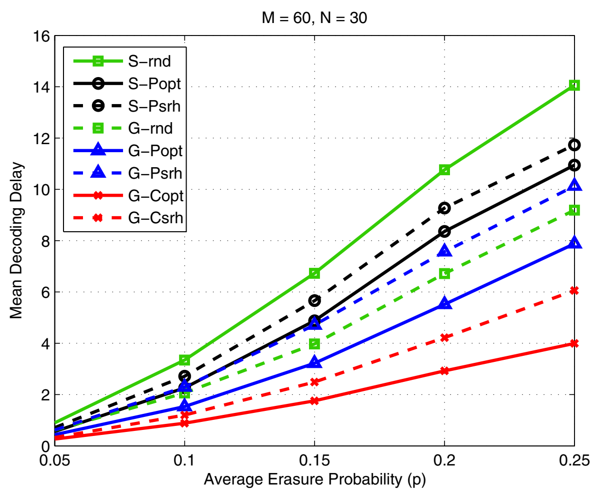

Fig. 3. Comparison of Mean Decoding Delays vs p

Moreover, it outperforms S-rnd, S-Psrh, G-rnd and G-Psrh by $60-67\%$, $38-62\%$, $35-45\%$ and $28-52\%$, respectively.

Figure 2 depicts the comparison of mean decoding delays achieved by the different algorithms against N, for $M = 60$ and $p = 0.15$. The figure shows that our proposed G-Copt algorithm outperforms the S-Popt and G-Popt algorithms by $61-63\%$ and $38-45\%$, respectively. It also shows that our proposed G-Csrh algorithm outperforms the optimal S-Popt and G-Popt algorithms by $41-55\%$ and $12-29\%$, respectively. Moreover, it outperforms S-rnd, S-Psrh, G-rnd and G-Psrh by $59-66\%$, $48-60\%$, $33-45\%$ and $38-59\%$, respectively.

Figure 3 depicts the comparison of mean decoding delays achieved by the different algorithms against p, for $M = 60$ and $N = 30$. The figure shows that our proposed G-Copt algorithm outperforms the S-Popt and G-Popt algorithms by $53-64\%$ and $40-49\%$, respectively. It also shows that our proposed G-Csrh algorithm outperforms the optimal S-Popt and G-Popt algorithms by $41-45\%$ and $21-24\%$, respectively. Moreover, it outperforms S-rnd, S-Psrh, G-rnd and G-Psrh by $57-64\%$, $48-56\%$, $34-46\%$ and $40-48\%$, respectively.

We can finally see that the performance gap, between our proposed optimal and heuristic algorithms, increases with the increase of N and p, but not M. In fact, the larger N and/or p, the larger the number of vertices of each receiver in $\mathcal{G}$, the larger the number of non-connected vertices added to $\mathcal{G}$ (as vertices of the same receiver are not connected), the larger the number of maximal cliques in it, the lower the efficiency of the maximum weight vertex search algorithm to track the optimal maximum weight clique. This is not the case for the increase of M, as it does not change the level of vertex connectivity.

VII. Conclusion

In this paper, we first introduced the concept of G-IDNC to further minimize decoding delay in wireless broadcast, compared to S-IDNC. G-IDNC loosens the strict instant decodability constraint in [1], [2] in order to target more receivers, while preserving the attractive properties of S-IDNC. We showed that the minimum decoding delay problem for G-IDNC can be formulated as a maximum weight clique problem, with reception success probabilities as vertex weights. We then designed a simple heuristic G-IDNC algorithm based on maximum weight vertex search, which tracks the optimal performance with increasing degradation for large frame sizes and average erasure probabilities. However, simulation results show that our proposed optimal and heuristic algorithms outperform several optimal and heuristic approaches for both S-IDNC and G-IDNC.

References

[1] P. Sadeghi, D. Traskov, and R. Koetter, “Adaptive network coding for broadcast channels,” *Fifth Workshop on Network Coding, Theory and Applications (NetCod’09)*, pp. 1–6, June 2009.

[2] P. Sadeghi, R. Shams, and D. Traskov, “An optimal adaptive network coding scheme for minimizing decoding delay in broadcast erasure channels,” *EURASIP Journal of Wireless Communations and Networking*, pp. 1–14, April 2010.

[3] J. Sundararajan, D. Shah, and M. Medard, “Online network coding for optimal throughput and delay - the three-receiver case,” *International Symposium on Information Theory and Its Applications (ISITA’08)*, pp. 1–6, Dec. 2008.

[4] L. Keller, E. Drinea, and C. Fragouli, “Online broadcasting with network coding,” *Fourth Workshop on Network Coding, Theory and Applications (NetCod’08)*, pp. 1–6, Jan. 2008.

[5] E. Drinea, C. Fragouli, and L. Keller, “Delay with network coding and feedback,” *IEEE International Symposium on Information Theory (ISIT’09)*, pp. 844–848, June 2009.

[6] S. El Rouayheb, M. Chaudhry, and A. Sprintson, “On the minimum number of transmissions in single-hop wireless coding networks,” *IEEE Information Theory Workshop (ITW’07)*, pp. 120–125, Sept. 2007.

[7] M. Chaudhry and A. Sprintson, “Efficient algorithms for index coding,” *IEEE Conference on Computer Communications Workshops (INFOCOM’08)*, pp. 1–4, April 2008.

[8] M. Garey and D. Johonson, *Computers and Intractability - A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

[9] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer. Berlin, 1999.

[10] K. Yamaguchi and S. Masuda, “A new exact algorithm for the maximum weight clique problem,” *23rd International Conference on Circuits/Systems, Computers and Communictions (ITC-CSCC’08)*, 2008.