

# Completion Delay Reduction in Lossy Feedback Scenarios for 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 study the effect of packet feedback loss events on the broadcast completion delay performance of instantly decodable network coding. These feedback loss events result in a continuous lack of knowledge about the reception status at different subsets of receivers. This lack of knowledge creates a challenge in selecting efficient packet combinations in subsequent transmissions. To solve this problem, we first identify the different possibilities of unheard feedback events at the sender and determine their probabilities. Given these probabilities and the nature of the problem, we design three partially blind instantly decodable network coding approaches that perform coding decisions similar to the algorithms proposed in [1], [2], but on blindly updated graphs to account for unheard feedback events. These three approaches are then compared through extensive simulations. Results show that re-considering all the attempted packet requests, with unheard feedback, in subsequent coding decisions can achieve a tolerable degradation against the perfect feedback performance for relatively high feedback loss probabilities.

**Index Terms**—Wireless Broadcast; Instantly Decodable Network Coding; Lossy Feedback.

## I. INTRODUCTION

Network coding applications in packet transmission and recovery over wireless erasure channels have recently attracted much attention [1], [3]–[7]. In [1], [6], [7], an important subclass of network coding, namely the instantly decodable network coding (IDNC), was considered due to its numerous desirable properties, such as instant packet recovery, simple XOR-based packet encoding and decoding, and no buffer requirements.

One major drawback of IDNC is that it is not a rate-optimal approach and thus may result in high completion delay and low throughput. In [1], [2], we studied the problem of minimizing the completion delay in IDNC and showed that finding its optimal solution is intractable. Nonetheless, we employed the problem properties and structure to design simple maximum weight clique search algorithms, which were shown to almost achieve the optimal completion delay performance in wireless multicast and broadcast scenarios. In [8], we extended our proposed algorithms to the case of intermittent feedback, in which feedback is received accurately at the sender but after several packet transmissions.

The proposed algorithms in [1], [2], [8] and most other opportunistic network coding works assume that the received feedback from all the receivers is perfect and is not subject

to losses. Although a high level of protection for feedback packets can be employed in several network settings, such as cellular and WiMAX systems, unavoidable occasions of deep fading over wireless channels can still expose them to loss events. Moreover, other network settings cannot guarantee the correct arrival of each feedback packet at the sender due to transmission power limitations and possible interference with other feedback.

In these lossy feedback scenarios, the sender will receive feedback packets from only a subset of the targeted receivers, after a given transmission, and thus the status of these receivers can be updated in the IDNC graph [2]. For the other targeted receivers whose feedback is not heard at the sender, the latest status of packet reception and requests will be unknown. Consequently, the sender must blindly estimate the status of these receivers, in order to perform the subsequent IDNC transmission. In this following transmission, the sender may receive feedback packets from some of these receivers but will lose the feedback of others. Consequently, the sender must continuously perform partially blind IDNC decisions until a correct completion feedback is received from all the receivers.

In this paper, we address the following question: *How can we extend our proposed IDNC algorithms to efficiently operate in lossy feedback scenarios?* To the best of our knowledge, this paper is a first step in studying the effect of feedback losses on the completion delay of IDNC. It both identifies the different possibilities of the sender’s uncertainties in events of unheard feedback and proposes some approaches to deal with these uncertainties so as to reduce the IDNC completion delay.

We first compute the probability mass function (pmf) of the receiver’s reception status, given an unheard feedback event from this receiver. Given this pmf and the properties of the completion delay problem, we design three partially blind instantly decodable network coding approaches that perform coding decisions according to the identified strategy in [1], [2] but using blindly updated IDNC graphs to account for unheard feedback events. We then test these three approaches of partially blind graph updates and compare their performance through extensive simulations.

The rest of the paper is organized as follows. In Section II, we introduce the system model and parameters. The IDNC graph is illustrated in Section III. In Section IV, we derive the pmf of the receiver’s reception status in lossy feedback

scenarios. Our proposed modified IDNC algorithms with the three different partially blind graph update approaches are introduced in Section V-D and their performance is compared 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 transmission phase*. Each sent packet is subject to loss (a.k.a. erasure) at receiver  $i$  with probability  $p_i$ , which is assumed to be fixed during the frame transmission period. Each receiver listens to all transmitted packets and feeds back to the sender a positive acknowledgement (ACK) for each received packet. By the end of the initial transmission 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  $i$ .
- The *Wants* set (denoted by  $\mathcal{W}_i$ ) is defined as the set of packets that are lost by receiver  $i$  in the initial transmission 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 *state feedback matrix (SFM)*  $\mathbf{F} = [f_{ij}]$ ,  $\forall i \in \mathcal{M}, j \in \mathcal{N}$ , such that  $f_{ij} = 0$  if  $j \in \mathcal{H}_i$  and  $f_{ij} = 1$  if  $j \in \mathcal{W}_i$ .

After the initial transmission phase, a recovery transmission phase starts. In this phase, the sender exploits the SFM to transmit XORed combinations of the source packets. We define the *targeted receivers* by a transmission as the receivers that can instantly decode the sent packet in this transmission to extract a new source packet. The non-targeted receivers that receive non-instantly decodable packets discard them. After each transmission, the targeted receivers, which receive the coded packet, send ACK packets that are used by the sender to update the SFM. Note that this condition implies that a targeted receiver, which lost the sender's transmission, will not generate a feedback since it would not know it was originally targeted. This process is repeated until all receivers declare that they obtained all the packets. We define the *completion delay* of a frame as the number of recovery transmissions required until all receivers obtain all the packets.

To be fair in comparison with the original perfect feedback formulation and algorithms in [1], [2], in terms of feedback frequency, we assume that a receiver does not send any feedback unless it is targeted by a packet. In other words, if a feedback is lost by one of the targeted receivers, the sender will not get any feedback from this receiver until the next transmission in which it is targeted. We also assume that each feedback sent from a receiver includes acknowledgements of all previously received packets. Finally, we assume channel reciprocity, which means that the packet loss probabilities seen by any receiver  $i$ , on both forward (sender to receiver) and reverse (receiver to sender) links, are the same and are both equal to  $p_i$ .

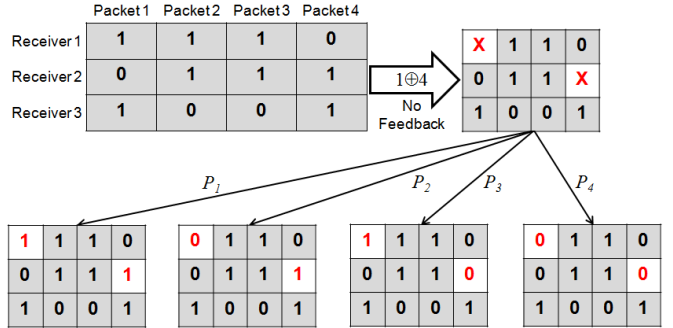


Fig. 1. Illustration of the potential uncertainty in lossy feedback scenarios

## III. IDNC GRAPH

The IDNC graph is a graph that defines the set of all feasible instantly decodable packet combinations. It was first introduced in the context of a heuristic algorithm design solving the index coding problem [9], [10] and was extended to IDNC in [11]. The IDNC graph  $\mathcal{G}$  is constructed by first generating a vertex  $v_{ij}$  in  $\mathcal{G}$  for each packet  $j \in \mathcal{W}_i, \forall i \in \mathcal{M}$ . Two vertices  $v_{ij}$  and  $v_{kl}$  in  $\mathcal{G}$  are adjacent if one of the following conditions is true:

- $j = l \Rightarrow$  The two vertices are induced by the loss of the same packet  $j$  by two different receivers  $i$  and  $k$ .
- $j \in \mathcal{H}_k$  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.

Consequently, each edge between two vertices in the graph represents a coding opportunity that is instantly decodable for the two receivers inducing these vertices. Given this graph formulation, we can easily define the set of all feasible packet combinations in IDNC as the set of packet combinations defined by all cliques in  $\mathcal{G}$ . Consequently, the sender can generate an IDNC packet for a transmission by XORing all the packets identified by the vertices of a selected clique in  $\mathcal{G}$ .

## IV. RECEPTION STATUS DISTRIBUTION

As previously mentioned, the presence of a probability of feedback loss creates uncertainties at the sender about the reception status of the targeted receivers with unheard feedback. In other words, the sender does not perfectly know the packets received at the different receivers so as to accurately determine subsequent instantly decodable coded packets. This notion of uncertainty is illustrated in Figure 1, in which the SFM is as shown on the top left corner and the sender sends the packet combination  $1 \oplus 4$  with the aim to deliver packets 1 and 4 to receivers 1 and 2, respectively. If the sender does not receive feedback from both receivers 1 and 2, it cannot concretely decide on whether to switch the entries  $f_{11}$  and  $f_{24}$  from 1 to 0 but would rather be uncertain about their status, as shown in the top right SFM. This uncertainty results in a conditional probability distribution (conditioned on the fact of unheard feedback) over the four SFMs shown at the bottom. Since the uncertainty on the values of  $f_{11}$  and  $f_{24}$  are independent from

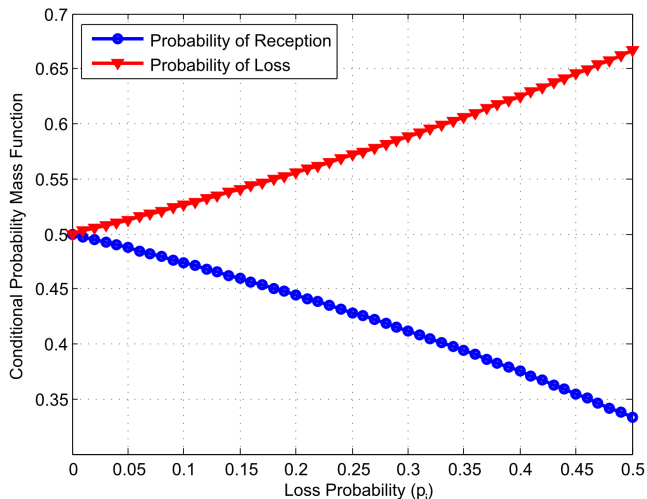


Fig. 2. Conditional pmf variation as a function of the loss probability  $p_i$

each other, we can focus on the reception status distribution for only one receiver.

To compute the conditional pmf of a receiver's reception status given an unheard feedback, we need to quantify the probabilities of the different cases causing unheard feedback events at the sender after a single transmission. Given the feedback model and the channel reciprocity condition explained in Section II, the unheard feedback event from a targeted receiver  $i$  could mean one of two sub-events:

- 1) The packet was not received by  $i$  and thus it did not issue a feedback. This event can occur with probability  $p_i$ .
- 2) The packet was received by  $i$ , and  $i$  issued a feedback packet that did not arrive at the sender. This event can occur with probability  $(1 - p_i)p_i$ .

If any of these two events occurs, and if the packet intended for receiver  $i$  is packet  $j$ , then the position  $f_{ij}$  in the feedback matrix will be uncertain. It can be equal to 1 (packet not received) if the first event occurred or 0 (packet received) if the second event occurred. Consequently, an unheard feedback event  $\mathbb{U}_i$  from the targeted receiver  $i$  with packet  $j$ , will render  $f_{ij}$  (which reflects the reception status of receiver  $i$ ) a random variable with Bernoulli pmf defined as:

$$\begin{aligned}
 P_{L|\mathbb{U}_i}^i &= \mathbb{P}(f_{ij} = 1 | \mathbb{U}_i) = \frac{p_i}{p_i + (1 - p_i)p_i} \\
 &= \frac{1}{2 - p_i}
 \end{aligned} \tag{1}$$

$$\begin{aligned}
 P_{R|\mathbb{U}_i}^i &= \mathbb{P}(f_{ij} = 0 | \mathbb{U}_i) = \frac{(1 - p_i)p_i}{p_i + (1 - p_i)p_i} \\
 &= \frac{1 - p_i}{2 - p_i}.
 \end{aligned} \tag{2}$$

Figure 2 depicts the variation in the conditional pmf of the reception status at a targeted receiver  $i$  as a function of its packet loss probability  $p_i$ , given an unheard feedback event.

It is clear from both the above two expressions and the figure

that, for any value of  $p_i$  and given an unheard feedback from a targeted receiver, the probability that this receiver has lost the packet is always greater than or equal to the probability that it has received it. In other words, given an unheard feedback event from a receiver, the estimation of the sender that this receiver did not receive the packet is the maximum likelihood estimation. Nonetheless, Figure 2 shows that this likelihood is not dominant. Even for a packet loss probability of 0.5 at a receiver, making a decision that this receiver did not receive the packet, given an unheard feedback event from it, can be wrong with probability 0.34. In the next section, we will employ the above observations to extend the operation of our designed IDNC completion delay reduction algorithms [1], [2] to lossy feedback scenarios.

## V. PARTIALLY BLIND IDNC ALGORITHMS

In [1], we showed that the IDNC completion delay in the perfect feedback problem is significantly reduced if the sender gives more priority to targeting the receivers with larger values of  $\psi_i \triangleq \frac{|W_i|}{1 - p_i}$ . This parameter represents the expected residual completion delay of receiver  $i$  if it is persistently targeted in all transmissions until completion. We will refer to it as *persistent residual completion delay (PRCD)*. This prioritization is implemented by assigning a weight  $\psi_i$  to each vertex  $v_{ij}$  in the IDNC graph [1]. In [2], we generalized the vertex weights to be  $\psi_i^n$ , where  $n$  determines the degree of bias given to receivers with higher PRCDs. The set of targeted receivers for each transmission can then be determined by running a maximum weight clique search over this weighted IDNC graph.

In the lossy feedback scenarios, the uncertainty in the reception status of different receivers, studied in Section IV, affects the ability of the sender to both determine the instant decodability conditions of coded packets at the different receivers and efficiently compute their PRCDs for prioritization. Clearly, this can greatly affect the IDNC completion delay. In other words, the sender cannot directly employ the designed algorithms in [1], [2] to efficiently reduce the IDNC completion delay. To solve this problem, we propose and compare three partially blind IDNC approaches that estimate the current reception status of all receivers, then apply our efficient algorithms on the corresponding blindly updated IDNC graph, to select the subsequent transmission's clique. Each of the three approaches focuses on one or more of the problem properties with the hope to achieve a lower completion delay.

### A. No Vertex Elimination (NVE)

In this approach, all the vertices in of a transmission, for which the sender did not hear a feedback, are all kept in the graph and are all considered for potential service in subsequent transmissions, until a feedback is received for them. According to the analysis in Section IV, this approach follows the maximum likelihood estimates for all uncertain vertices in subsequent transmissions. Moreover, these vertices (and thus their packet requests) will be rapidly re-attempted, thus giving the chance to the sender to receive feedback from

these receivers and to determine their accurate reception status. This approach will widely re-attempt a lot of vertices, which may slow down the steps towards completion. Moreover, Figure 2 shows that, even for a relatively high packet loss rate such as 0.5, the NVE approach may fall into many estimation errors and thus may result in worse performance.

### B. Full Vertex Elimination (FVE)

In this approach, all attempted vertices with unheard feedback, in each transmission, are eliminated from the graph. In case there are no remaining vertices in the graph while the system did not receive completion feedback from all receivers, the sender retransmits combinations of the remaining uncertain vertices. In the lossy feedback context, we can see that the FVE approach results in IDNC graphs including only the vertices that have never been attempted before. Consequently, FVE guarantees the innovation of all transmitted coded packets, regardless of the uncertainty in the partially observed system status. Nonetheless, this innovation comes at the cost of going against the maximum likelihood estimates for all uncertain vertices. Given these most likely wrong elimination decisions in FVE, the sender will always underestimate both the system state and the PRCDs of some receivers, which may result in prioritization errors and thus may degrade the completion delay performance.

### C. Stochastic Vertex Elimination (SVE)

In this approach, the attempted vertices with unheard feedback are eliminated from the graph probabilistically, according to the conditional reception probabilities of their inducing receivers. When a vertex is attempted in a transmission and no feedback is heard from its receiver, the sender keeps this vertex in the graph with probability  $P_{L|U_i}^i$  and eliminates it with probability  $P_{R|U_i}^i$ . In case there are no remaining vertices in the graph while the system did not receive completion feedback from all receivers, the sender retransmits combinations of the remaining uncertain vertices.

SVE tends to balance the properties of both NVE and FVE. Unlike NVE, SVE does not always keep the attempted vertices with unheard feedback but rather give some chance to their elimination, proportionally to their conditional reception probabilities. Thus, it tends to reduce the number of re-attempted vertices and gives more opportunity to transmitting new packets towards completion. On the other hand, SVE better represents the conditional reception probabilities of the receivers, compared to FVE, and thus can both re-attempt the non-received vertices and update the feedback matrix earlier.

### D. Algorithm Implementation

After obtaining the partially blind updated graph, using one of the approaches described in the previous sections, we can assign the weights  $\psi_i^n$  to each vertex  $v_{ij}$  in the graph and perform a maximum weight clique search algorithm, as proposed in [1], [2]. The maximum weight clique selection algorithm is known to be NP-hard but can be exactly solved in polynomial time for non-large graphs [12]. Nonetheless, this

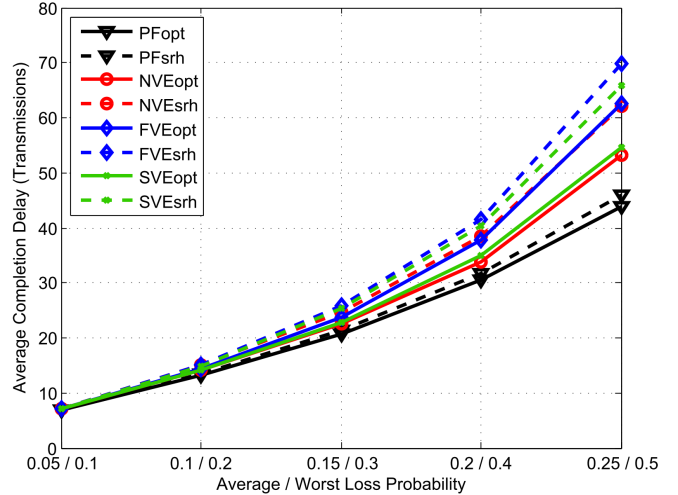


Fig. 3. Average completion delay vs  $p/p_w$

complexity may still be prohibitive in large networks. For these cases, we can employ the quadratic time maximum weight vertex search algorithm, also proposed in [1], [2]. In this algorithm, the weights of the vertices are defined as follows. Define  $a_{ij,kl}$  as the adjacency indicator of vertices  $v_{ij}$  and  $v_{kl}$  in  $\mathcal{G}$  such that  $a_{ij,kl} = 1$  if they are adjacent and is zero otherwise. We then define the vertex weight of vertex  $v_{ij}$  as:

$$w_{ij} = \psi_i^n \sum_{\forall v_{kl} \in \mathcal{G}} a_{ij,kl} \psi_k^n. \quad (3)$$

Thus, a large vertex weight reflects both its high PRCD and its adjacency to a large number of vertices belonging to receivers with large PRCDs. The cliques in this algorithm are thus built by sequentially selecting the maximum weight vertex from among the ones that are adjacent to all previously selected ones. In each step, the vertex weights are recomputed within this adjacent subgraph only, as explained in [1], [2].

## VI. SIMULATION RESULTS

In this section, we compare through extensive simulations the performance of the three partially blind graph update approaches, using both the maximum weight clique selection algorithm (denoted by “opt”) and the maximum weight vertex search algorithm (denoted by “srh”). We also compare the performance of these approaches to those of the perfect feedback (PF) IDNC algorithms, as a performance benchmark for IDNC. For all the above cases, we will employ the  $n = 3$  realization of our proposed algorithms due to its best performance in [2].

Figure 3 depicts the comparison of the average completion delay, achieved by the different algorithms, against the average and worst packet loss probabilities, for  $M = 60$  and  $N = 30$ . For each point  $(p/p_w)$  in the x-axis, the packet loss probabilities of different receivers are random variables with mean  $p$  and worst-case value of  $p_w$ . Figure 4 depicts the same comparisons against  $M$  for  $N = 30$ ,  $p = 0.15$  and  $p_w = 0.3$ .

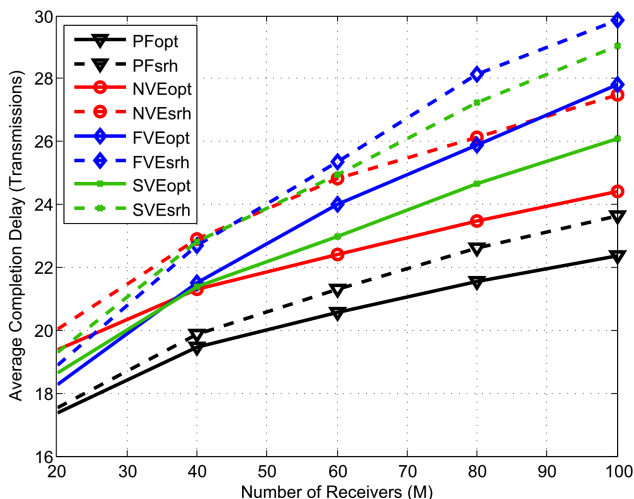


Fig. 4. Average completion delay vs  $M$

From Figure 3, we can see that NVE achieves the best performance for the entire range of loss probabilities. At high loss probabilities, the performance of FVE considerably deviates from that of NVE and SVE as, at these probabilities, assuming correct reception has a very high chance of error. Thus, FVE will be assigning wrong priorities to the receivers, which results in this considerable degradation.

Figure 4 shows that FVE outperforms the other techniques for very small receiver population whereas NVE dominates for more than 40 receivers. This can be explained in the light of the characteristics of the two approaches as follows. At low number of receivers, the number of vertices in the IDNC graph is relatively smaller compared to that at large numbers of receivers. Consequently, the time needed by FVE, to both attempt all these vertices and start to re-attempt unacknowledged vertices, is small. Consequently, FVE does not find the time to drift very far from the actual system state and to fall in prioritization errors. In this case, the packet innovation in FVE plays the role of achieving its better performance over NVE.

For large numbers of receivers, the larger size of the IDNC graph makes the time for FVE to attempt all vertices longer. Consequently, each receiver, whose last vertex or several vertices were attempted but unacknowledged, will have to wait longer for FVE to re-attempt them. This effect causes more prioritization errors and a longer drift from the actual state of the system, which greatly degrades the performance of FVE. On the other hand, NVE reduces these effects since it both better tracks the actual receivers' reception status and leaves the attempted vertices with unheard feedback in the graph, which increases the speed of their transmission re-attempt, recovery and feedback reception.

Finally, we can observe a degradation in the average completion delay obtained in the lossy feedback scenario compared to the perfect feedback scenario. However, for a relatively large network setting ( $M = 100$ ,  $N = 30$ ), a worst loss

probability of 0.3, and a broadcast setting, this degradation in the frame delivery duration (from the start of the frame transmission until its reception at all receivers) reaches 3.9% and 7.1% for NVEopt and NVEsrh, respectively, compared to the perfect feedback algorithm performance. These values are clearly tolerable in such large networks and up to 30% feedback loss probability, which is typically very high for signalling information.

## VII. CONCLUSION

In this paper, we studied the effect of packet feedback loss events on the broadcast completion delay of IDNC. To overcome the uncertainty arising in this lossy feedback environment, we first computed the conditional pmf of the different possibilities given unheard feedback events at the sender. Given this pmf and the nature of the problem, we designed and compared three partially blind instantly decodable network coding approaches that perform coding decisions similar to the algorithms proposed in [1], [2], but on blindly updated graphs to account for unheard feedback events. Simulation results show that the no-elimination of vertices can achieve both the best performance for a wide range of scenarios and a tolerable degradation against the perfect feedback performance for relatively high feedback loss probabilities.

## REFERENCES

- [1] S. Sorour and S. Valaee, "On minimizing broadcast completion delay for instantly decodable network coding," *IEEE International Conference on Communications (ICC'10)*, May 2010.
- [2] —, "Completion delay minimization for instantly decodable network coding," *submitted to IEEE/ACM Transactions on Networking*, pp. 1–12, Dec. 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, Jun. 2009.
- [6] 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 Communications and Networking*, pp. 1–14, Apr. 2010.
- [7] S. Sorour and S. Valaee, "Minimum broadcast decoding delay for generalized instantly decodable network coding," *IEEE Global Telecommunications Conference (GLOBECOM'10)*, 2010.
- [8] —, "Completion delay minimization for instantly decodable network coding with limited feedback," *accepted for publication in IEEE International Conference on Communications (ICC'11)*, 2011.
- [9] M. Chaudhry and A. Sprintson, "Efficient algorithms for index coding," *IEEE Conference on Computer Communications Workshops (INFOCOM'08)*, pp. 1–4, Apr. 2008.
- [10] 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, Sep. 2007.
- [11] S. Sorour and S. Valaee, "Adaptive network coded retransmission scheme for wireless multicast," *IEEE International Symposium on Information Theory (ISIT'09)*, pp. 2577 – 2581, Jun. 2009.
- [12] K. Yamaguchi and S. Masuda, "A new exact algorithm for the maximum weight clique problem," *23rd International Conference on Circuits/Systems, Computers and Communications (ITC-CSCC'08)*, 2008.