Effect of Feedback Loss on 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 probabilistic and prolonged packet feedback loss events on the broadcast completion time of instantly decodable network coding (IDNC). These feedback loss events result in a lack of knowledge about the reception status at different subsets of receivers, which creates a challenge in selecting efficient IDNC packet combinations in subsequent transmissions. To solve this problem for both probabilistic and prolonged feedback loss, we first identify the different possibilities of feedback loss events at the sender and determine their probabilities in both cases. Given these probabilities and the nature of the IDNC completion time problem, we design three 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 feedback events. These three approaches are then compared through extensive simulations. Results show that the full consideration and the full negligence of all the attempted packet requests with probabilistic and prolonged feedback loss events, respectively, in subsequent coding decisions can achieve a tolerable degradation against the perfect feedback performance for relatively high feedback loss probabilities and periods.**

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

I. Introduction

The application of network coding in packet transmission and recovery over wireless erasure channels have attracted much attention in the past few years. [1], [3]–[8]. In [1], [6]–[8], an important subclass of NC was introduced under the name of instantly (or instantaneously) decodable network coding (IDNC), in which the received network coded packets must be decoded at their reception instants and cannot be stored for future decoding. IDNC was considered in these works due to its practicality and numerous desirable properties, such as instant packet recovery, simple XOR-based packet encoding and decoding, and no buffer requirements. In IDNC, the sender must select the packet combination in each network coded packet transmission according to the previously received packets at all receivers, so that this coded packet is instantly decodable at a specific set of receivers if not all of them. The selection of the appropriate packet combinations that are instantly decodable at specific sets or all the receivers is done through what is known as the IDNC graph [1], [2], [8]–[10].

One major drawback of IDNC is that it is not a rate-optimal approach and thus may result in high completion time and low throughput. In [1], [2], we studied the problem of minimizing the completion time 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 time performance in wireless multicast and broadcast scenarios.

The proposed algorithms in [1], [2] and most other opportunistic network coding works assume that the received feedback from all the receivers is perfect and is not subject to loss. This assumption has its limitations in many practical scenarios because feedback loss events may occur at the sender due to several practical settings and impairments in wireless networks. One reason for such feedback loss events is probabilistic erasures due to channel impairments on the uplink channels from the receivers. Although a high level of protection for feedback packets can be employed in several networks, such as cellular and WiMAX systems, fast fading effects over wireless channels can still expose them to probabilistic loss events. Moreover, other network settings cannot guarantee the correct arrival of each feedback packet at the sender due to transmission power limitation and possible interference with other feedback.

In these probabilistic feedback loss 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.

Another type of feedback loss at the sender may be more prolonged, such that all the feedback from one or all receivers is not received at the sender for several subsequent transmissions, due to the correlation in channel impairments or to the setting of the network. One example may be one or multiple receivers may be in prolonged deep fading conditions due to shadowing. Another very practical example is the case of time division duplex (TDD) modes in cellular and WiMAX

978-1-4577-9538-2/11/$26.00 ©2011 IEEE

systems. In this network setting, the sender must transmit multiple packets in the downlink frame without receiving any feedback until the uplink frame starts. In this sense, we may consider the sender as loosing all feedback information from all the receivers. In all these scenarios, the sender must make blind IDNC decisions during the feedback loss period. These blind decisions, made without any knowledge of the reception status at different receivers, will definitely affect the IDNC completion time.

All the aforementioned scenarios raise the following question: *How can we extend our proposed IDNC algorithms to efficiently operate in probabilistic and prolonged feedback loss scenarios?* In this paper, we will answer the above question for the reciprocal erasure channel example as a probabilistic feedback loss scenario and the TDD mode example as a prolonged feedback loss scenario. Nonetheless, the designed algorithms for these examples can be easily extended to all other examples in both scenarios. We first identify the probability mass functions (pmfs) of the receiver's reception status, given an unheard feedback event from this receiver in both probabilistic and prolonged feedback loss scenarios. Given these pmf properties and the nature of the completion time problem, we design three 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 compare these three approaches of blind graph updates for the probabilistic and prolonged feedback scenarios 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. Our previous work on completion time reduction in IDNC is summarized in Section IV. In Section V, we derive the pmfs of the receiver's reception status in both feedback loss scenarios. Our proposed modified IDNC selection algorithms with the three different blind graph update approaches are introduced in Section VI-B and their performance is compared in Section VII. Finally, Section VIII 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 feedbacks to the sender a positive acknowledgement (ACK) for each received packet. At the end of the initial transmission phase, three sets of packets are attributed to each receiver i: 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 network coded combinations of the source packets. We define the *targeted receivers* by a transmission as the receivers that can instantly decode a packet from this transmissions. The non-targeted receivers that receive non-instantly decodable packets discard them. After each transmission, the targeted receivers that decoded a packet send ACK packets, which 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 feedback that they obtained all their requested packets. We define the *completion time* of a frame as the number of transmissions required from the start of the frame transmission until all receivers obtain all their intended packets. Given the considered model, the completion time has a fixed section of N transmissions in which all frame packets are transmitted uncoded and a variable section in which IDNC recovery transmissions are sent until all receivers obtain all their intended receivers. We refer to the latter section as *completion delay*.

Let p and p_w be the average and worst packet erasure probabilities of all receivers, respectively. In the probabilistic feedback loss scenario, we assume channel reciprocity, which means that the packet erasure 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. Similar to the schemes proposed in [1], [2], we assume in both scenarios that the 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 in both scenarios that each feedback sent from a receiver includes acknowledgements of all previously received packets.

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 [11], [12] and was first extended to IDNC in [9]. 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.

Feedback Matrix

	Packet 1	Packet 2	Packet 3	Packet 4
Receiver 1	1	1	1	0
Receiver 2	0	1	1	1
Receiver 3	1	0	0	1

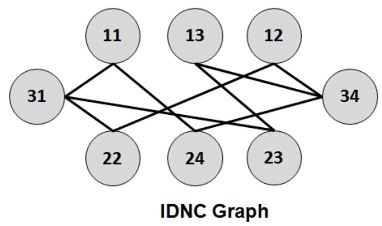

Fig. 1. Example of a feedback table and its corresponding IDNC graph

- $j \in \mathcal{H}_k$ 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 $\mathcal{G}$.

Figure 1 depicts an example of a feedback table and its corresponding IDNC graph. In this example, if the clique consisting of the vertices v_{11} and v_{24} is selected in the IDNC graph, then sending packet $1 \oplus 4$ will help both receiver 1 and 2 to recover packet 1 and 4, respectively, as each of them has the other packet and can use XOR to decode their missing packet out of this sent coded packet.

IV. IDNC Completion Time Reduction with Perfect Feedback

In [1], [2], we formulated the problem of minimizing the completion delay problem as a stochastic shortest path (SSP) problem. The exponentially growing dimensions of this formulation with the number of receivers and packets made its solution intractable. Nonetheless, we were able to employ the properties of this SSP to show that the completion time can be efficiently reduced by any policy that can both bring the system closest to absorption in each step and maximizes the density of coding opportunities in the IDNC graph.

By studying the geometric structure of the SSP in [2], we showed that the policy, which gives more priority to targeting the receivers with larger values of ψ_i^n, such that:

$$\psi_i \triangleq \frac{|\mathcal{W}_i|}{1-p_i} \ , \tag{1}$$

efficiently brings the system closest to the absorption state. The parameter ψ_i represents the expected remaining number of transmissions required by receiver i until it receives all its missing packets, assuming that it is persistently targeted in all transmissions until completion. We will refer to it as the *persistent residual completion delay (PRCD)* of receiver i. The exponent n determines the degree of bias given to receivers with larger PRCDs. In [2], [10], we showed that the same policy also efficiently densifies the coding opportunities in the IDNC graph.

Given this result, we designed an IDNC algorithm that implements the above prioritization by assigning a weight ψ_i^n to each vertex v_{ij} in the IDNC graph. The set of targeted receivers for each transmission can then be determined by running a maximum weight clique search over this weighted IDNC graph. However, the maximum weight clique selection algorithm is known to be NP-hard but can be exactly solved in polynomial time for non-large graphs [13]. Nonetheless, this complexity may still be prohibitive in large networks. For these cases, we designed a quadratic time maximum weight vertex search algorithm [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

$$a_{ij,kl} = \begin{cases} 1 & v_{ij} \text{ is adjacent to } v_{kl} \text{ in } \mathcal{G}(s) \\ 0 & \text{otherwise .} \end{cases} \tag{2}$$

We then define the vertex weight of vertex v_{ij} as:

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

Thus, a large vertex weight reflects both its high PRCD and its adjacency to a large number of vertices belonging to receivers with large PICDs. 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 weights of the adjacent vertices are recomputed within this adjacent subgraph only, as explained in [1], [2].

Figures 2 and 3 depict the average completion time performance of our proposed algorithms, for $n = 3$, $n = 5$ and $n = 10$ (denoted by L_3, L_5 and L_{10}, respectively), against the number of receivers M (for $N = 30$, $p = 0.15$, $p_w = 0.3$) and the average/worst erasure probability p/p_w (for $\mu = 0.5$ and 1, $M = 60$, $N = 30$), respectively. In both figures, we compare the performance of our proposed optimal maximum weight clique selection (denoted by "opt") to that of our proposed maximum weight vertex search (denoted by "srh") algorithm, as well as the maximum clique (MC) selection algorithm and full network coding (FNC). In this FNC scheme, we assume that all generated coding coefficients are always linearly independent and thus this perfect FNC scheme achieves the optimal completion time performance over all network coding schemes [14].

From both figures, we can see that the algorithm tends to converge to the same performance with the smallest completion time achieved by the L_3 and L_5 algorithms. For L_{10}, the performance slightly degrades. We can also observe that the heuristic maximum weight vertex search algorithms perform

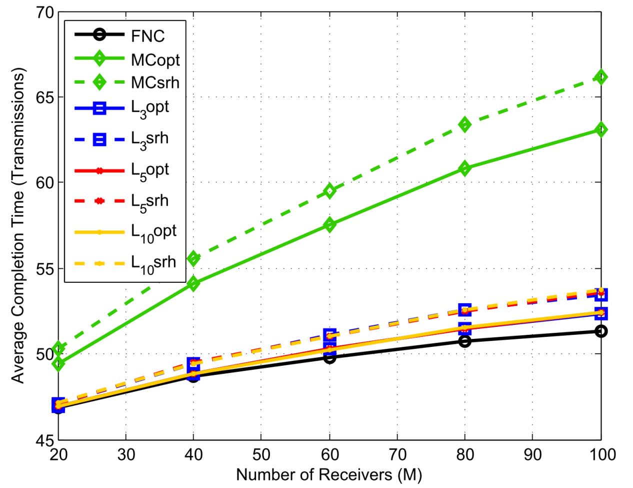

Fig. 2. Average completion time for the perfect feedback scenario vs M

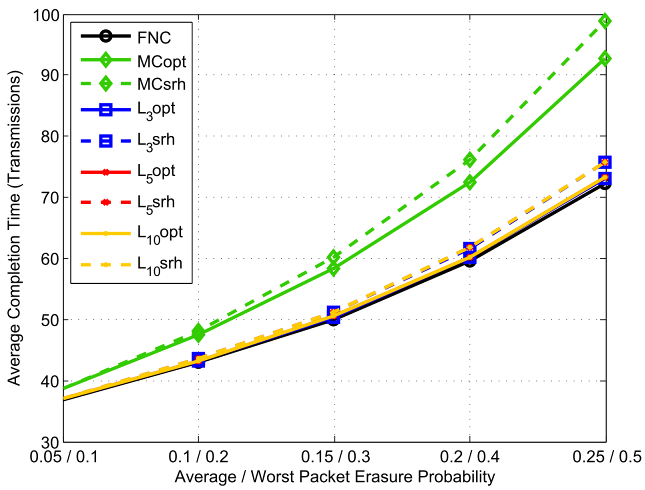

Fig. 3. Average completion time for the perfect feedback scenario vs p/p_w

very closely to the optimal clique selection algorithms for all values of n. For a relatively large network setting ($M = 100$ and $N = 30$) and relatively harsh channel conditions ($p = 0.15$ and $p_w = 0.3$), the proposed heuristic achieves 2.2% degradation. Moreover, our proposed maximum weight clique selection and maximum vertex search algorithms with values of n considerably outperforms the maximum clique selection algorithms in terms of average completion time. Finally, results show that our proposed algorithms almost achieves the optimal performance of the perfect FNC scheme. For a relatively large network setting ($M = 100$ and $N = 30$) and relatively harsh channel conditions ($p = 0.15$ and $p_w = 0.3$), the proposed optimal and heuristic clique selection algorithms achieve a degradation of 1.3% and 3.6%, respectively, against the optimal completion time performance achieved by the perfect FNC scheme. This near-optimal performance is achieved while fully preserving the important and practical benefits of INDC

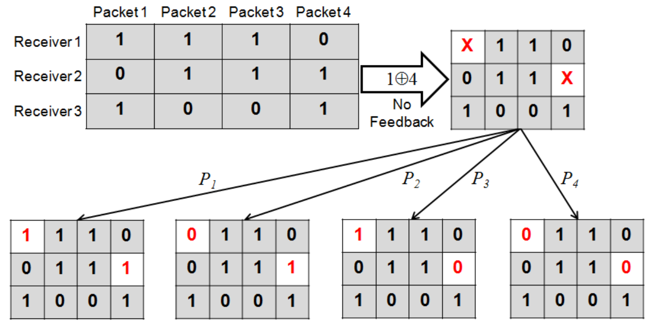

Fig. 4. Illustration of the potential uncertainty in lossy feedback scenarios, given the SFM in Figure 1 and after sending packet $1 \oplus 4$

compared to FNC.

V. Reception Status Distribution

A. Reciprocal Probabilistic Feedback Loss Scenario

As previously mentioned, the possibility of having feedback loss events create uncertainty at the sender about the reception status of the different receivers. 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 4, in which the SFM of Figure 1 is 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, each representing a combination of receiving or not receiving for each of the two receivers. Since the uncertainty on the values of f_{11} and f_{24} are independent from 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 event, 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

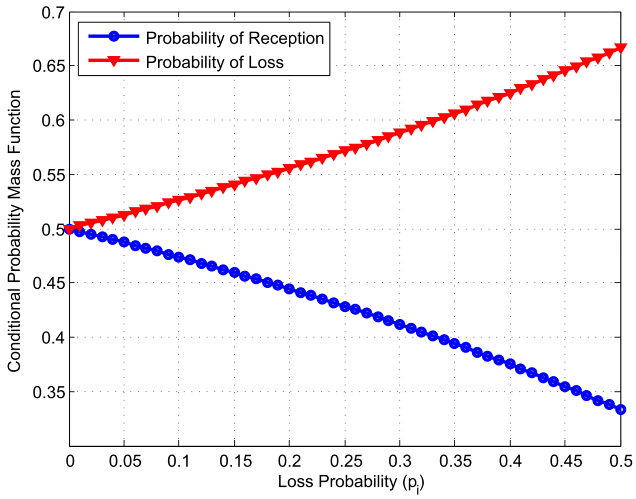

Fig. 5. Conditional pmf variation as a function of the erasure probability p_i

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:

$$P^i_{L|\mathbb{U}_i} = \mathbb{P}\left(f_{ij} = 1\middle|\mathbb{U}_i\right) = \frac{p_i}{p_i + (1-p_i)p_i} = \frac{1}{2-p_i} \tag{4}$$

$$P^i_{R|\mathbb{U}_i} = \mathbb{P}\left(f_{ij} = 0\middle|\mathbb{U}_i\right) = \frac{(1-p_i)p_i}{p_i + (1-p_i)p_i} = \frac{1-p_i}{2-p_i}\ . \tag{5}$$

Figure 5 depicts the variation in the conditional pmf of the reception status at a targeted receiver i as a function of its packet erasure 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 lost the packet is always greater than or equal to the probability that it 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 5 shows that this likelihood is not dominant. Even for a packet erasure 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 time reduction algorithms [1], [2] to lossy feedback scenarios.

B. *TDD Prolonged Feedback Loss Scenario*

Unlike the probabilistic feedback loss scenario, the probability of not hearing a feedback from all the receivers is 1. When a packet is sent from the sender, the probabilities of it being received and lost at receiver i, given that no feedback can be heard, are $(1-p_i)$ and p_i, respectively. Consequently, the likelihood of reception status of any receiver depends only on its packet erasure probability. Since in most communication systems, it is more likely that the packet erasure probability is less than 0.5, the estimation of the sender that the receivers received a transmitted packet is the maximum likelihood estimation.

VI. Blind IDNC Algorithms

In both probabilistic and prolonged feedback loss scenarios, the uncertainty in the reception status of different receivers, studied in Section V, affects the ability of the sender to both certainly 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 time. In other words, the sender cannot directly employ the designed algorithms in [1], [2], summarized in Section IV, to efficiently reduce the IDNC completion time. To solve this problem, we propose and compare three partially blind IDNC approaches that blindly estimate the current reception status of all receivers, then apply our efficient algorithms on the corresponding blindly updated IDNC graph to select the cliques for the subsequent transmissions. Each of the three approaches focuses on one or more of the problem properties with the hope to achieve a lower completion time.

A. *Blind IDNC Graph Update*

1) *No Vertex Elimination (NVE):*

In this approach, all the vertices in the IDNC graph, representing the served packet requests (represented by vertices) with unheard feedback, are not removed from the graph and are all kept in subsequent transmissions. These vertices (and thus their packet requests) will be rapidly re-attempted in this scenario, thus giving the chance to the sender to receive feedback from these receivers and to determine their accurate reception status. Nonetheless, this approach will widely re-attempt a lot of vertices which, may slow down the steps of the process towards completion.

According to the analysis in Section V-A, this approach follows the maximum likelihood estimates of the system status in the case of probabilistic feedback loss scenario. However, Figure 5 shows that this likelihood is not always dominant. Even for a relatively high packet loss rate such as 0.5, the NVE approach may fall into many estimation errors. The effect of this trade-off will be illustrated in Section VII-A.

For the prolonged feedback loss scenario, the NVE approach is totally opposite to the maximum likelihood estimation of the receivers' reception status and thus is expected to achieve a very high degradation.

2) *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 the 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 an IDNC graph with only certain vertices that are not attempted before. Consequently, using FVE guarantees innovation in all generated packets regardless of the uncertainty in the current reception status. For the probabilistic feedback loss scenario, this innovation of FVE comes at the cost of going against the maximum likelihood estimates for all uncertain vertices. This most likely wrong estimation not only deviates the system from its true state but also reduces the perceived PICDs of the involved receivers at the sender, which may result in prioritization errors and thus may degrade the completion time performance.

For the prolonged feedback loss scenario, this approach follows the maximum likelihood estimation of the system status for packet erasure probabilities less than 0.5. For higher packet erasure probability, FVE will not be the maximum likelihood estimation of the system. Nonetheless, the innovative approach of FVE may still make it achieve a good performance. The effect of this trade-off will be illustrated in Section VII-B.

3) Stochastic Vertex Elimination (SVE): In this approach, the attempted vertices with unheard feedback are eliminated from the graph probabilistically, according to the reception probabilities of their inducing receivers. In case of the probabilistic feedback loss scenario, 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 $P^i_{L|\mathbb{U}_i}$ and eliminates it with probability $P^i_{R|\mathbb{U}_i}$. For the prolonged feedback loss scenario, the probabilities of keeping and eliminating the unacknowledged vertices are p_i and $(1-p_i)$, respectively. In case there are no remaining vertices in the graph while the system did not receive the 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 gives 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 in an earlier stage and update the feedback matrix earlier. This compromise between the NVE and SVE properties may or may not succeed in reducing the IDNC completion time, compared to both approaches, and is thus interesting to consider and test.

B. Algorithm Implementation

After obtaining the blind updated graph, using one of the approaches described in the previous sections, we can assign the weights ψ^n_i to each vertex v_{ij} in the graph and perform a maximum weight clique search algorithm or a maximum weight vertex search algorithm, as explained in Section IV.

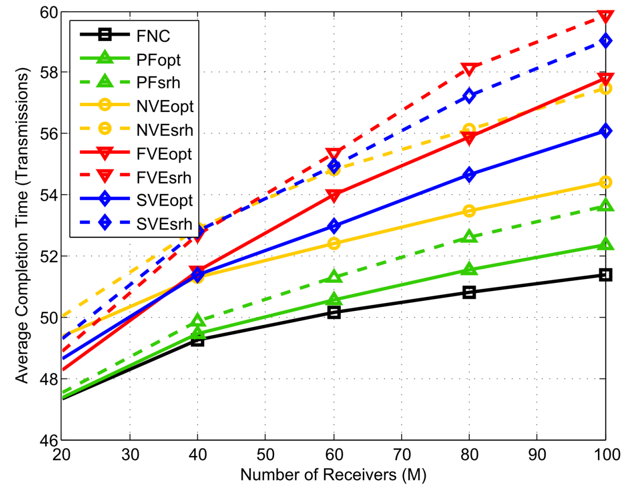

Fig. 6. Average completion time for the probabilistic feedback loss scenario vs M

VII. 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 good performance shown in Figures 2 and 3.

A. Probabilistic Feedback Loss Scenario

Figure 6 depicts the comparison of the average completion time achieved by the different algorithms against M for $N = 30$, $p = 0.15$ and $p_w = 0.3$. Figure 7 depicts the same comparisons against the average and worst packet erasure probabilities, for $M = 60$ and $N = 30$.

From Figure 7, we can see that NVE achieves the best performance for the entire range of packet erasure probabilities. At high erasure probabilities, the performance of FVE considerably deviates from that of NVE and SVE because assuming correct reception, at these probabilities, has a very high chance of error. Thus, FVE will be assigning wrong priorities to the receivers, which results in this considerable degradation.

Figure 6 shows that FVE outperforms the other approaches for very small numbers of receivers whereas NVE dominates for more than 40 receivers. In both ranges, SVE achieves an intermediate performance between the best and worst approaches. This can be explained in the light of the characteristics of the three approaches as follows. At low numbers

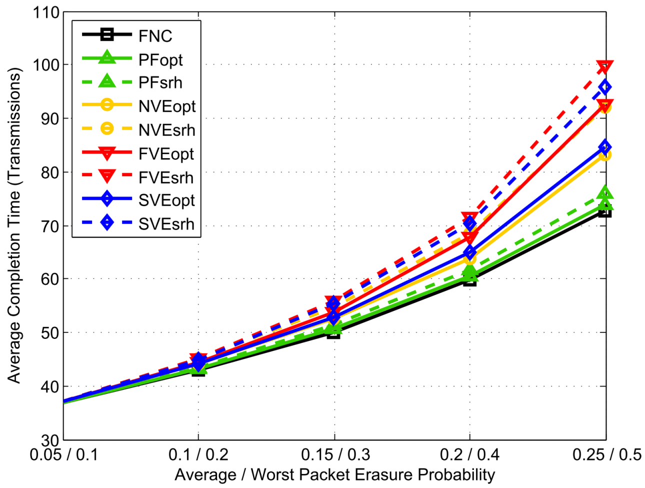

Fig. 7. Average completion time for the probabilistic feedback loss scenario vs p/p_w

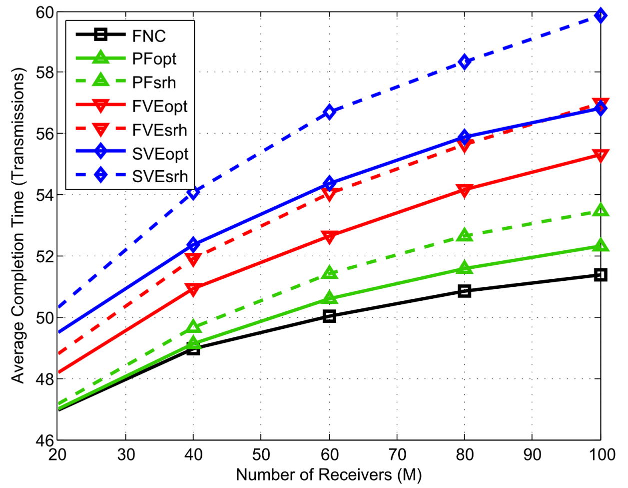

Fig. 8. Average completion time for the prolonged feedback loss scenario vs M

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. Thus, 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. This innovation property in SVE, being less than FVE and more than NVE, makes it perform worse than the former and better than the latter for this range of numbers of receives.

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 larger 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. For SVE, these faster re-attempt and better maximum likelihood tracking properties, being less than NVE and more than FVE, makes it perform worse than the former and better than the latter for this range of number of receivers.

Finally, we can observe a degradation in the average completion time 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 erasure probability of 0.3, and a broadcast setting, this degradation in the frame completion time reaches 3.9% and 7.1% for NVEopt and NVEsrh, respectively, compared to the perfect feedback algorithm performance. These two algorithm also achieve a degradation of 5.8% and 11.6%, respectively, against FNC, which achieves the optimal completion time performance over all network coding schemes. These degradation percentages are clearly tolerable in such very large network and up to 30% feedback loss probability, which is typically very high for signalling information.

B. Prolonged Feedback Loss Scenario

Figure 8 depicts the average completion time of the three proposed blind IDNC algorithms with that of the prompt feedback, against M, for $N = 30$, $T_f = 5$, $p = 0.15$ and $p_w = 0.3$. Figure 9 depicts the completion time comparison against T_f, for $M = 60$, $N = 30$ and $\mu = 0.5$. Finally, Figure 10 depicts the average completion time comparison against the packet erasure probability for $M = 60$, $N = 30$, $T_f = 5$ and $\mu = 10.5$. In Figure 10, we assume that all the receivers have the same packet erasure probability, which is equal to the corresponding value on the x-axis for each simulation point.

From all three figures, we can see that the FVE approach achieves the best completion delay performance compared to the other three approaches for both the optimal and search clique selection algorithms. This clearly shows that the different trade-offs introduced in the other three approaches tend to degrade the performance rather than improving it.

Another important observation is the degradation in average completion delay obtained in the limited feedback scenario compared to the full feedback scenario, which naturally increases with the increase of the feedback period. However, for a relatively large network setting ($M = 100$, $N = 30$) and a considerable feedback period value ($T_f = 5$), this degradation reaches 3 and 3.5 transmissions for the FVEopt and FVEsrh algorithms, respectively, compared to their corresponding full feedback algorithms (depicted in Figure 8). This results in an overall degradation in the completion time of 6% and 6.5% for FVEopt and FVEsrh, respectively. These two algorithms also

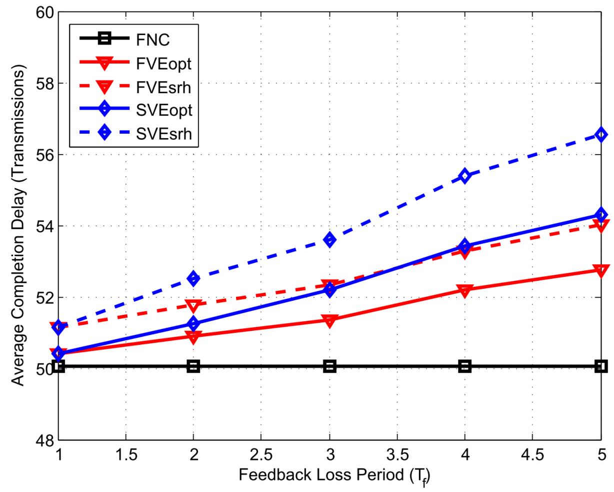

Fig. 9. Average completion time for the prolonged feedback loss scenario vs T_f

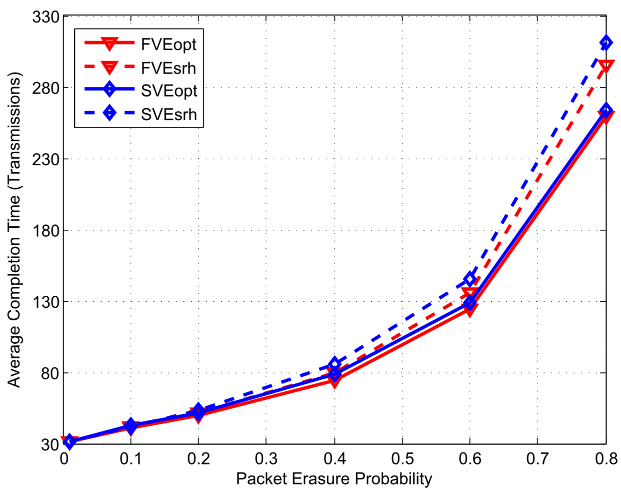

Fig. 10. Average completion time for the prolonged feedback loss scenario vs p

achieve a degradation of 7.7% and 9.7%, respectively, against the optimal completion time performance achieved by FNC. These degradation percentages are clearly tolerable given the 80% reduction in the feedback frequency, a reduction that is of extreme importance in many practical network settings as explained in Section I.

Finally, we observe from Figure 10 that FVE still outperforms SVE even for high erasure probabilities, where re-attempting the vertices earlier would have been intuitively better. The reason for this performance of FVE is both the innovation of packets within the feedback period and the re-attempt of non-received vertices after feedback instants.

VIII. Conclusion

In this paper, we studied the effects of probabilistic and prolonged feedback loss events on the broadcast completion time of IDNC. We first identified the different possibilities of feedback loss events at the sender and determined their probabilities in both probabilistic and prolonged packet feedback loss scenarios. Given these probabilities and the nature of the IDNC completion time problem, we designed three 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 feedback events. These three approaches are then tested through extensive simulations. For the probabilistic feedback loss scenario, results show that the no vertex elimination approach, which keeps all uncertain vertices in the graph and considers them in subsequent coding decisions, achieves both the best performance compared to the other approaches and a tolerable degradation, against the perfect feedback performance, for relatively high feedback erasure probabilities. In contrast, the full vertex elimination approach, which removes all uncertain vertices from the graph and ignores them in subsequent coding decisions, both outperforms the other approaches in the prolonged feedback loss scenario and can achieve a tolerable degradation for relatively large feedback loss periods.

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] D. Traskov, M. Medard, P. Sadeghi, and R. Koetter, “Joint scheduling and instantaneously decodable network coding,” *Global Telecommunications Conference (GLOBECOM’09)*, pp. 1 –6, Dec. 2009.

[7] 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, Apr. 2010.

[8] S. Sorour and S. Valaee, “Minimum broadcast decoding delay for generalized instantly decodable network coding,” *IEEE Global Telecommunications Conference (GLOBECOM’10)*, 2010.

[9] ——, “Adaptive network coded retransmission scheme for wireless multicast,” *IEEE International Symposium on Information Theory (ISIT’09)*, pp. 2577 – 2581, Jun. 2009.

[10] ——, “Coding opportunity densification strategies for instantly decodable network coding,” *submitted to IEEE Transactions on Communications*, pp. 1–12, Jan. 2011.

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

[12] 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.

[13] 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.

[14] D. Lun, M. Médard, R. Koetter, and M. Effros, “On coding for reliable communication over packet networks,” *Physical Communications*, vol. 1, no. 1, pp. 3–20, Mar. 2008.