On The Capacity of Immediately-Decodable Coding Schemes for Wireless Stored-video Broadcast with Hard Deadline Constraints

Xiaohang Li, Chih-Chun Wang, and Xiaojun Lin

***Abstract*—Multimedia streaming applications have stringent Quality-of-Service (QoS) requirements. Typically, each packet is associated with a packet delivery deadline. This work models and considers streaming broadcast of stored video over the downlink of a single cell. We first generalize the existing class of immediately-decodable network coding (IDNC) schemes to take into account the deadline constraints. The performance analysis of IDNC schemes are significantly complicated by the packet deadline constraints (from the application layer) and the immediate-decodability requirement (from the network layer). Despite this difficulty, we prove that for independent channels, the IDNC schemes are asymptotically throughput-optimal subject to the deadline constraints when there are no more than three users and when the video file size is sufficiently large. The deadline-constrained throughput gain of IDNC schemes over non-coding scheme is also explicitly quantified. Numerical results show that IDNC schemes strictly outperform the non-coding scheme not only in the asymptotic regime of large files but also for small files. Our results show that the IDNC schemes do not suffer from the substantial decoding delay that is inherent to existing generation-based network coding protocols.**

***Index Terms*—network coding, broadcast cellular networks, video streaming, delay/deadline-constrained systems, network capacity analysis, stochastic processing networks**

I. Introduction

The advance of broadband wireless technologies has enabled a number of innovative wireless services. Among them, video streaming over wireless networks has gained a significant amount of interest. In this paper, we are interested in streaming *stored-video* wirelessly to multiple receivers, where the video file is assumed to be available on the server at the very beginning of the transmission. Stored-video broadcasting is useful for applications such as collaborative learning and instant replay in a live sport event [1]. Note that in video streaming, each packet has a delivery deadline, which is sequentially placed along the time horizon (e.g., the first frame's deadline is at 1/30 second, while the second frame's deadline is at 2/30 second, and so on). If a packet is not delivered before the deadline, it is considered useless to the receivers. Unfortunately, the random and unreliable wireless channel makes it much more difficult to meet the deadline constraints of video packets, while maintaining a high system throughput at the same time. In this paper, we will focus on using network coding (NC) to improve the deadline-constrained streaming throughput over a one-hop wireless broadcast channel.

It is well-known that without deadline constraints, network coding (NC) can increase the throughput of communication networks [2], [3], [4], and can be efficiently implemented [5], [6]. NC is particularly attractive for unreliable wireless broadcast channels: when a packet needs to be retransmitted, coded retransmission is more efficient than uncoded retransmission because the coded packet can be made innovative[1] to all receivers rather than only a subset of the receivers [8]. On the other hand, NC also introduces "decoding delay," i.e., the receiver may not be able to decode the information packet right away. For example, in generation-based NC schemes [6], the receiver must accumulate a sufficient number of coded packets from a generation before it can decode any information packet. Such a long decoding delay can be detrimental to delay-sensitive applications such as video streaming. Hence, how to design a NC scheme that satisfies the deadline constraints becomes a challenging problem.

Existing studies have focused on various types[2] of delay in NC protocols. We note however that these studies do not apply to the stored-video streaming application under unreliable broadcast channels. Specifically, [9] discusses how different methods of encoding can affect the decoding delay while considering only noise-free channels. [10] studies the completion time for the entire file. In contrast, in the setting of video streaming an individual deadline needs to be imposed for *every* packet. [8], [11], [12], [13] study the average decoding delay or the queue-length growth rate when broadcasting packets over a wireless channel. However, for video streaming, meeting the deadline constraint is more critical than reducing the average delay. [7] studies the problem of minimizing the average/maximum decoding delay in the setting of multiple-description codes, and [14] proposes a dynamic coding-window-selection policy that optimizes deadline-constrained flow throughput. Nonetheless, they both focus on the setting where a set of packets have the same deadline, which is different from the sequential deadline setting of this paper.

Manuscript received 1, July, 2010, revised 6 December, 2010. This work was supported in parts by NSF grants CCF-0845968, CNS-0643145, CNS-0721484, CNS-0813000, CNS-0905331. Part of this paper was presented at the IEEE Conference on Computer Communications (INFOCOM), San Diego, USA, 2010.

X. Li, C.-C. Wang, and X. Lin are with the Center for Wireless Systems and Applications (CWSA), the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA (e-mail: {li179, chihw, linx}@purdue.edu).

[1]A coded packet is innovative to a user if this coded packet can bring new information to this user. For linear NC, an innovative packet increases the rank of the decoding matrix by one [6], [7].

[2]There are different types of delay, including queuing delay, propagation delay, decoding delay, and total transmission delay (see Fig. 1)

To combat the delay inefficiency of NC, recent practical protocols have focused on a new class of "immediately decodable" NC (IDNC) schemes [15], [16], [17]. For example, suppose two destinations Rx_1 and Rx_2 are interested in different packets X and Y, respectively, and also suppose that Rx_1 has overheard Y and Rx_2 has overheard X due to random channel realization. By carefully exploiting the feedback information, the base station can send $X + Y$, which is *immediately decodable* from both Rx_1 and Rx_2's perspectives. Compared to generation-based solutions, IDNC schemes (i) have substantially smaller decoding delay, (ii) require much smaller buffers to store the not-yet-decoded packets since the decoded packets can be expurgated from the buffer immediately, (iii) incur much lower encoding complexity since only the binary field is used, and (iv) incorporate naturally the feedback provided by existing ARQ mechanisms and is adaptive to the underlying channel realization. As a result, IDNC schemes generally demonstrate much faster startup phase [18], and is more suitable for time-sensitive applications.

In this work, we are interested in the achievable throughput of IDNC schemes under the sequential deadline constraints of stored-video streaming. Unfortunately, the performance analysis of IDNC schemes turns out to be highly non-trivial. Note that the coding decision of an IDNC scheme requires the to-be-coded packets in the backlog to satisfy certain patterns, which has some similarity to the constraints in stochastic processing networks [19]. It is well-known that such constraints lead to more complicated design and analysis than that for standard communication networks [19] even without deadline constraints. Prior studies of similar IDNC schemes either do not consider deadline-constraints at all [20], or only provide simulations but no analysis [18]. Moreover, most existing results only consider the simplest setting of two users, and have not explored the more intricate dynamics when coding over > 2 users (see Section IV-D for further discussion). To the best of our knowledge, there have been no analytical studies in the literature that analyze the throughput of IDNC schemes subject to sequential deadline constraints, despite the inherent simplicity and attractive numerical performance of IDNC protocols.

The main contribution of this paper is to provide such an analytical study. Specifically, we show that over an unreliable wireless broadcast channel, the IDNC schemes indeed achieve asymptotically the optimal throughput subject to deadline constraints, as the file size becomes large. In this analysis we use a novel form of Lyapunov functions, which reveals new and intricate dynamics of IDNC systems. We establish such results for the cases of 2 users and 3 users, respectively. As readers will see, the 2-user and 3-user cases already uncover non-trivial and interesting insights that could serve as a precursor to the full analysis for the case of an arbitrary number of users. These results are in sharp contrast to the existing observations that the throughput improvement of NC must come at the expense of longer delay. We also quantify analytically the coding gain over the non-coding policies. Our numerical simulations show that the throughput of the IDNC scheme has an almost instant start-up phase and is near throughput-optimal even for small file sizes.

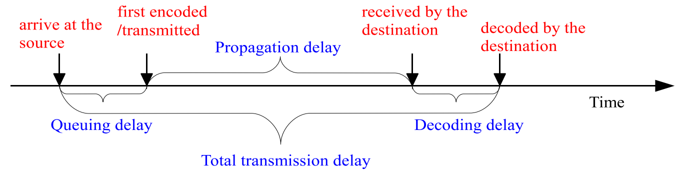

Fig. 1: Illustration of various aspects of delays

The rest of this paper is organized as follows. Section II introduces the system model. Section III describes the IDNC schemes for deadline-constrained streaming. Section IV provides the throughput analysis of IDNC schemes for up to 3 users, which is the main contribution of this paper. Section V provides an analytical capacity expression for the non-coding schemes and conducts numerical comparison between IDNC and non-coding schemes. Section VI concludes the paper.

II. System Model

We consider the downlink of a single cell in which the base station (BS) broadcasts a video file of N packets to M users. We define the time when the BS begins transmitting the first packet as the time origin, and we assume that all packets are available on the video-file server at the time origin. We assume that time is slotted. Each packet $n = 1, 2, \ldots, N$ has a deadline d_n, after which the packet is no longer useful for any of the M users. We assume that the deadline of the n-th packet is of the form

$$d_n = \lambda n \text{ , where } \lambda \text{ is a fixed positive integer.}$$

Each packet has to be delivered before its deadline d_n. The sequential deadlines model the scenario where the video frames must be played at a steady rate, e.g. every $1/30$ seconds.

We consider random and unreliable wireless channels. That is, a broadcast packet may be received by all users, a subset of users, or no users at all. Suppose a packet is transmitted in the t-th time slot. For $j = 1, 2, \ldots, M$, we use $C_j(t) = 1$ to denote that user j can receive the packet successfully, and $C_j(t) = 0$, otherwise. We consider the models in which channels are independently and identically distributed (i.i.d.) but may be spatially dependent. Specifically, the vector $(C_1(t), \cdots, C_M(t))$ is i.i.d. for all t, the marginal success probability is $\mathsf{P}(C_j(t) = 1) = p$ for all j, but $C_i(t)$ and $C_j(t)$ may be independent or not. We assume that at the end of each time slot, the BS has perfect feedback as to whether the packet has been successfully received by each user, based on which the decision of what to be sent in the next time slot will be made. The same feedback assumption has also been made in [7], [8], [11], [12], [18], [20], [21].

If coding is not allowed, the source can only transmit uncoded packets. Suppose packet n is transmitted at time t, and only a subset of users have received packet n successfully. After receiving the feedback at the end of time t, the BS may decide to retransmit the same packet n for other users that have not received packet n yet, or may decide to move to the next packet $n + 1$ to enhance the chance that packet $n + 1$ can be received before its deadline. If coding across different

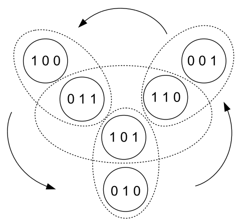

Fig. 2: Illustration of all six lists (L_{100}, L_{011}, L_{001}, L_{110}, L_{010}, and L_{101}) and all coding groups for the three-user case

N	Total number of packets
$n(t)$ or n	The index of the next uncoded packet to send at the end of time t
$q(t)$	The index advancement defined by $n(t) - \frac{t}{\lambda}$
τ_N	The first time slot when $n(t)$ becomes N
$\mathsf{RC}_j(t_1, t_2)$	Number of remaining coding opportunities involving user j at the end of time t_2, when only counting packets with index $\geq n(t_1)$
$Q(t_1, t_2)$	Defined by $\min_j \mathsf{RC}_j(t_1, t_2)$
$C_j(t_1, t_2)$	Number of channel successes for user j between time $(t_1, t_2]$
$L_{\mathbf{v}}$	The list of packets with receiving status $\mathbf{v}$

Fig. 3: Some key notations used in this work

packets is allowed, then in one slot, the BS can encode a set of unexpired packets together and broadcast it to all users. When coding is used, we require that an information packet is "decoded" before the corresponding deadline.

Our goal is to design a coding/scheduling policy that maximizes the number of successful (unexpired) packet receptions. Let $D_j(n) = 1$ if user j can successfully decode/recover the n-th information packet before its deadline $d_n = \lambda n$; and $D_j(n) = 0$, otherwise. We define the total number of successes N_{success} as $\sum_{n=1}^{N} \sum_{j=1}^{M} D_j(n)$. Our goal is to maximize the normalized expected throughput $\frac{\mathsf{E}\{N_{\text{success}}\}}{MN}$.

A. An Upper Bound on the Optimal Throughput

We note that the total number of packets that all users can recover/decode is upper bounded by the total number of channel successes. Therefore, $\mathsf{E}\{N_{\text{success}}\} \leq \mathbb{E}\left\{\sum_{t=1}^{\lambda N} \sum_{j=1}^{M} C_j(t)\right\} = M\lambda N p$. Further, since the best scenario is that each user can recover/decode all N information packets, we have $\mathsf{E}\{N_{\text{success}}\} \leq MN$. Jointly, we have

$$\frac{\mathsf{E}\{N_{\text{success}}\}}{MN} \leq \min(\lambda p, 1). \tag{1}$$

We next propose an IDNC scheme that asymptotically achieve the above upper bound when N tends to infinity.

III. IDNC Schemes for Deadline-Constrained Systems

The main focus of this work is on the immediately decodable network coding (IDNC) schemes. That is, whenever a receiver receives a coded packet, we require that such receiver can immediately decode one more information packet. The requirement of immediate decodability imposes constraints on the set of packets that can be mixed together. Specifically, we define the (binary) receiving status vector $\text{Rec}(n)$ for packet n at a given time as $(D_1(n)\, D_2(n)\, \ldots\, D_M(n))$. At the end of the t-th slot, we define $L_{\mathbf{v}}$ as the list of unexpired packets such that the receiving status $\text{Rec}(n)$ is $\mathbf{v}$, where $\mathbf{v}$ is an M-dimensional binary vector that is neither all-zero nor all-one.[3] More explicitly, $L_{\mathbf{v}} \triangleq \{n : \lambda n > t, \text{Rec}(n) = \mathbf{v}\}$. These lists are maintained at the BS. For any set of K ($2 \leq K \leq M$) packets, n_1 to n_K, the receiving status vectors form a $M \times K$ matrix $(\text{Rec}(n_1)^T, \text{Rec}(n_2)^T, \ldots, \text{Rec}(n_K)^T)$. If for every row of this matrix, the integer sum of all elements in this row is exactly $K-1$, these K packets can be added together, and form an immediately-decodable network-coded (IDNC) packet. Note that by this definition, each user has successfully received/decoded $K-1$ of these K packets already. Hence, if these K packets are mixed together and broadcast, every user who successfully receives the coded packet can *immediately decode* the remaining packet that it has not received before. If packets from different lists can form an IDNC packet, then these lists are said to constitute a coding group. A coding group is called non-empty if all lists in this coding group are non-empty. Or equivalently, we say that we can form an IDNC packet when there is a non-empty coding group. Take the case of $M = 3$ for example. There are 6 different lists, L_{001} to L_{110}, and totally they form 4 coding groups, which are illustrated by the big dashed circles in Fig. 2.

[3] We do not consider $L_{11\ldots1}$ and $L_{00\ldots0}$ since $L_{11\ldots1}$ is the list of packets that have been received by all users, while $L_{00\ldots0}$ is the list of packets that have not been received by any users.

A. Description of The IDNC Policies

The IDNC scheme is described by the following pseudo-code in which t denotes the time slot and n denotes the index of the next uncoded packet that the BS can transmit.

Set $n \leftarrow 1$, set all $L_{\mathbf{v}} \leftarrow \emptyset$, for all $\mathbf{v} \neq (0\ldots0)$ and $\mathbf{v} \neq (1\ldots1)$.
for $t = 1$ to λN **do**
  In the beginning of the t-th time slot, do the following:
  if $n \leq N$ **then**
    if there exists at least one non-empty coding group **then**
      Choose one non-empty coding group; generate and broadcast an IDNC coded packet .
    else
      Send uncoded packet n directly.
    end if
  else
    Choose the oldest packet i in all $L_{\mathbf{v}}$, and send packet i uncodedly.
  end if
  In the end of the t-th time slot,
  if we sent an uncoded packet in time t, and it is received by at least one user **then**
    $n \leftarrow n + 1$.

16:     **end if**
17:     UPDATE all $L_{\mathbf{v}}$, based on the feedback received from all users.
18: **end for**

The subroutine "UPDATE all $L_{\mathbf{v}}$" in Line 17 is described as follows: After a coded packet is transmitted, the receiving status of all packets involved may change. Suppose the receiving/decoding status for packet n changes from $\mathbf{v}$ to $\mathbf{v}'$, then we remove packet n from $L_{\mathbf{v}}$ and add it to $L_{\mathbf{v}'}$. If it is an uncoded packet that is transmitted and suppose its receiving status is $\text{Rec}(n) = \mathbf{v}$ that is neither all-zero nor all-one, then we add packet n to the list $L_{\mathbf{v}}$. During the update, we also expurgate all expired packets (with indices $\leq \frac{t}{\lambda}$).

One critical step of the algorithm is Line 6. Note that when $M = 2$, there are only two lists L_{01} and L_{10}, and there is only one coding group, which consists of these two lists. Hence, the BS just needs to check whether L_{01} and L_{10} are both nonempty [21]. If so, then the BS picks one packet from each list in the coding group and mixes them by binary XOR. One common choice is to select the oldest[4] such packet from each list. Although the algorithm is simple for $M = 2$, the situation becomes much more complicated when the number of users grows, because for a given time t, there may be multiple non-empty coding groups. Line 6 thus needs to choose one non-empty group among many. For example, the BS can choose the coding group with the smallest number of constituent lists, or simply choose the coding group under some probability distribution. Note that after deciding the coding group, the BS still has the freedom to decide which packets from the lists should be mixed together as was discussed in the $M = 2$ case. For the following, we will use the term "*generic* IDNC scheme" to refer to the IDNC scheme that does not specify the policy how to choose the coding group and how to choose which packets to be coded together.

In Section IV-B we will provide some key propositions that could be useful for analyzing any generic IDNC scheme. Then in Sections IV-C and IV-D, we analyze and prove the asymptotic throughput optimality of some IDNC schemes with specific policies of choosing the coding group/packets for the cases of $M = 2$ and 3, respectively. In general, rigorous capacity analysis often depends on how we choose the coding group/packets and is still an open problem for the case of $M > 3$.

Remark 1: A unique feature of the IDNC schemes is that it is *universal* in the sense that it does not require prior knowledge of the channel success probability p, which makes it very appealing for practical applications.

Remark 2: After n reach N, there may exist further opportunities to form IDNC packets. However, such opportunities may not always exist. For convenience, we only consider uncoded transmission in Line 11. In Section IV, we can prove the throughput optimality even without considering the transmission after n reaches N.

[4]The oldest packet is the one with the smallest index, while the youngest packet has the largest index.

IV. Performance Analysis for IDNC Schemes

This section contains a two-step analysis that proves the optimality of certain IDNC schemes for $M = 2$ and 3. More specifically, we first provide a sufficient condition (Lemma 1, Propositions 2 and 3) for a generic IDNC scheme to be asymptotically throughput optimal subject to hard deadline constraints. This condition is sufficient for an arbitrary number of users. Then we show that such a sufficient condition holds naturally when $M = 2$ and prove that the sufficient condition also holds for certain IDNC scheme when $M = 3$. We believe that our analysis also sheds important insights towards proving the optimality of IDNC schemes for general M values.

For convenience to the reader, we have summarized in Fig. 3 several key notations used in this section.

A. High-Level Ideas for Throughput-Optimality with Deadlines

The analysis of IDNC schemes with deadlines is complicated by the following two aspects. First, to form an IDNC packet, one must be able to find packets from multiple lists that form some patterns, which is similar to the constraints in stochastic processing networks [19] and thus substantially complicates the analysis. Second, a packet may be removed from a list if it has expired. Also, packets involved in the coded transmission may join different lists after the coding operation. Therefore, the model in [19] does not seem to apply to our case.

In this work, we provide a deadline-constrained throughput analysis based on a new Lyapunov function. To motivate our choice of the Lyapunov function, it is worthwhile to understand in what situation an IDNC scheme may potentially perform poorly. Recall that as long as $n \leq N$, an IDNC scheme will never send a coded packet that only benefits a subset of the users, nor will it *retransmit* an old uncoded packet (until it runs out of new uncoded packets, see Line 11). Under the setting of infinite backlog and no deadline constraints, this property guarantees throughput optimality as each packet is always serving all M users [18]. However, with deadlines, what could happen is that during the operation of the protocol, the BS might not have *enough* opportunities to transmit coded packets due to expiration. More explicitly, an IDNC scheme initially will keep transmitting new uncoded packets if all coding groups are empty. However, when the overhearing pattern of the latest packet n finally matches that of another packet n' so that they form an IDNC packet (note that $n' < n$), the older packet n' might have already expired and been removed from the list. In the extreme case, when an IDNC scheme finally encounters some non-empty coding groups, a significant fraction of packets might have expired and the successful throughput will degrade significantly.

In order to capture this effect, we introduce the quantity τ_N, which is the first time slot when the variable n in the proposed IDNC scheme becomes N (i.e., the file size). Note that during the interval $[1, \tau_N]$, the BS either transmits an uncoded packet that is new to all users, or transmits a coded packet that is innovative to, and immediately decodable by, all users. However, during the interval $(\tau_N, \lambda N]$, i.e., after τ_N

and before the last packet expires, the BS will have to transmit some coded packets that are innovative to only a subset of the users, which degrades the throughput of the system. For ease of exposition, suppose first that $p = \frac{1}{\lambda} - \epsilon$, for some small $\epsilon > 0$. Consider two extreme scenarios. If the BS found very few opportunities to form coded packets during the interval $[1, \tau_N]$, then τ_N could be as small as $N/(1-(1-p)^M)$ (because the index n could have been advanced by 1 as long as one of the users received the uncoded transmission). In this case, the loss of throughput in the interval $[\tau_N, N]$ will be quite significant. On the other hand, if τ_N is close to λN, it implies that the BS had found many opportunities to form coded packets, which "slows down" the advancement of n. In this case, because the expected reward for each time slot before τ_N is Mp, the total expected reward during $[1, \tau_N]$ is thus $Mp\tau_N \approx MNp\lambda$, which already approaches the capacity upper bound in (1). Fortunately, our analysis below shows that the latter scenario is indeed the one that is more likely to occur (at least for large N), given some mild conditions of the underlying channel model.

Based on the above observation, let $n(t)$ denote the value of the variable n at the end of time slot t, which is the index of the next uncoded packet that the BS will send. Define the "index advancement" at time t as: $q(t) \overset{\Delta}{=} n(t) - \frac{t}{\lambda}$. Note that if $q(t)$ remains finite with high probability when $N \to \infty$, then $q(\tau_N) = n(\tau_N) - \tau_N/\lambda$ is small. Note that by definition of τ_N, we have $n(\tau_N) = N$. Therefore, the condition that $q(\tau_N) = N - \tau_N/\lambda$ is finite implies that $\tau_N \approx \lambda N$ for large N. In the following, we consider the asymptotic regime of $N \to \infty$ (the file size becomes very large) and use Lyapunov stability to prove that $q(t)$ is finite/stable for $t \in [1, \infty)$ with probability one (for the case of $p < 1/\lambda$).

B. Key Propositions for Asymptotic Throughput Optimality

We begin with a lemma that holds for an arbitrary number of users. Let $C_j(t_1, t_2) = \sum_{t=t_1+1}^{t_2} C_j(t)$ denote the number of time slots in $(t_1, t_2]$ in which the transmitted packets are successfully received by user j. Note that in every time slot, a packet, (either coded or uncoded), is transmitted to all users. Next we introduce the notion of a coding opportunity involving a particular user. Note that when we mix packets from a non-empty coding group and transmit an IDNC packet, for any user j only one of the constituent lists $L_{\mathbf{v}}$ will have the j-th bit of receiving status $\mathbf{v}$ being 0. Namely, the packet in that list $L_{\mathbf{v}}$ has been received/decoded by some other users but not by user j, and the content of that packet can be decoded by user j if the IDNC packet arrives user j successfully. Due to this reason, at any time t (before mixing packets), we say that packet n is a (potential) *coding opportunity* involving user j when the j-th bit of the receiving status of packet n is zero (and not all bits are zero). Being a coding opportunity, packet n can later be combined with other packets to form an IDNC packet. The number of coding opportunities involving user j is then defined as the summation of the sizes of all $L_{\mathbf{v}}$ for which the j-th bit of $\mathbf{v}$ is zero (recall that we never consider the all-zero $\mathbf{v}$). For example, for the three-user case $M = 3$, the sum of the number of coding opportunities involving user 2 is $|L_{100}| + |L_{101}| + |L_{101}|$. Note that the number of coding opportunities evolves over time as the sizes of the lists $L_{\mathbf{v}}$ change due to packet reception and/or due to packet expiration. We can then prove the following lemma.

Lemma 1. *For any two time slots satisfying $t_1 < t_2$, if $t_2 < \lambda n(t_1)$, then*

$$n(t_2) - n(t_1) \leq \max_{j=1,\cdots,M} C_j(t_1, t_2) + Q(t_1, t_2),$$

where $Q(t_1, t_2) = \min_{j=1,\cdots,M} \mathsf{RC}_j(t_1, t_2)$, and $\mathsf{RC}_j(t_1, t_2)$ (which stands for "Remaining Coding opportunities") is the number of packets with index $n \geq n(t_1)$ that are coding opportunities involving user j at the end of time t_2. Namely, we only count the uncoded packets that are transmitted during the interval $(t_1, t_2]$.

Lemma 1 is a central result of this work. It upper bounds "how fast" the index n of a generic IDNC scheme can grow by relating it to the number of channel successes and a critical term $Q(t_1, t_2)$. Hence it is critical to proving τ_N grows at the rate of λN as discussed in Section IV-A. The next proposition shows that if $p < \frac{1}{\lambda}$, and $Q(t_1, t_2)$ satisfies a probabilistic condition (2), then $q(t)$ has a negative drift whenever $q(t)$ is large.

Proposition 2. *Consider the case $p < \frac{1}{\lambda}$. If for any $\epsilon'_1 > 0$, there exists a $B_1 > 0$ such that for all $B > B_1$ and $t_1 > 0$*

$$\mathsf{P}(Q(t_1, t_1 + B) \geq \epsilon'_1 \lambda B | q(t_1) > B) < \epsilon'_1, \tag{2}$$

then

$$\mathsf{E}\{q(t_1 + \lambda B) - q(t_1) | q(t_1) > B\} < \epsilon_2 B$$

for some $\epsilon_2 > 0$.

The proofs for Lemma 1 and Proposition 2 can be found in Appendices A and B, respectively. Proposition 2 shows that if $p < 1/\lambda$, and condition (2) holds, then $q(t)$ has a negative drift. Next we show that the negative drift of $q(t)$ is a sufficient condition for the asymptotic throughput optimality of an IDNC scheme when $p < 1/\lambda$.

Proposition 3. *Consider the case of $p < 1/\lambda$. If for all $B > B_1$ and $t_1 > 0$*

$$\mathsf{E}\{q(t_1 + \lambda B) - q(t_1) | q(t_1) > B\} < \epsilon_2 B$$

for some $\epsilon_2 > 0$, then $\lim_{N\to\infty} \frac{\mathsf{E}\{N_{success}\}}{MN} = \lambda p$.

Proof: Following classic steps of the Lyapunov-condition-based analysis, we can first show that the negative drift of $q(t)$ implies that for any $\epsilon, \epsilon' > 0$, there exists an $t_0 > 0$ such that $\mathsf{P}(q(t) < \epsilon' t) > 1 - \epsilon$, for $t > t_0$ (see [22] for more details). By plugging in the definition of $q(t)$, we thus have

$$1 - \epsilon < \mathsf{P}(n(t) - t/\lambda < \epsilon' t) = \mathsf{P}\left(\frac{Mpt}{Mn(t)} > \frac{p\lambda}{1 + \lambda\epsilon'}\right). \tag{3}$$

for all $t > t_0$, where the equality of (3) follows from simple arithmetic rearrangement.

Recall that the IDNC scheme guarantees that each of the t transmissions can serve all M users. There are totally $n(t) -$

1 packets that participate in the first t time slots. Therefore, $Mpt/(M(n(t)-1))$ is the normalized throughput for the first $n(t)-1$ packets. Equation (3) implies that the normalized throughput satisfies

$$\lim_{t\to\infty} \mathsf{E}\left\{\frac{Mpt}{M(n(t)-1)}\right\} \geq \frac{p\lambda}{1+\lambda\epsilon'}(1-\epsilon). \tag{4}$$

We then notice that when $t=\tau_N$, we have $n(t)-1=N-1$ and $Mp\tau_N$ is the expected total throughput for all users (i.e., N_{success}). By choosing arbitrarily small ϵ' and ϵ, we have proven that $\lim_{N\to\infty}\frac{\mathsf{E}\{N_{\text{success}}\}}{MN}\geq\lambda p$. By the upper bound in Section II-A, we also have that for the case of $p<1/\lambda$, $\frac{\mathsf{E}\{N_{\text{success}}\}}{MN}\leq\lambda p$. The result of Proposition 3 then follows. ■

Remark 1: Lemma 1, Propositions 2 and 3 together convert the problem of proving the asymptotic throughput optimality of an IDNC scheme for the case of arbitrary M users to the problem of proving whether the quantity $Q(t_1,t_2)$ satisfies (2). In the following, we will see that $Q(t_1,t_2)$ satisfies (2) trivially for the case of $M=2$, and we will prove that $Q(t_1,t_2)$ satisfies (2) for the case of $M=3$ for certain IDNC scheme.

Remark 2: In the symmetric channel setting, maximizing the normalized overall throughput also achieves perfect fairness. For example, when $p<1/\lambda$, when the IDNC schemes achieve the optimal throughput, the expected throughput for each user is $p\lambda N$, which is also optimal for the individual user.

Remark 3: Thus far, we have considered only the case of $p<\frac{1}{\lambda}$. The case of $p\geq\frac{1}{\lambda}$ can be derived by similar techniques and the details are shown in [22].

C. The Two-User Case

For $M=2$, whenever both lists L_{01} and L_{10} are nonempty, the BS would begin to send coded packets. If the coded packet is received by any one of the users, then the participating packet(s) will be decoded and removed from the lists. Only when one of the two lists is empty, say $|L_{10}|=0$, the BS will send an uncoded packet, which may later increase the length of the empty list L_{10} by at most one. From the above reasoning, at any time instant, one of the two lists L_{10} and L_{01} must have at most one packet. That is, there exists a user j for which the number of coding opportunities involving user j is one. Hence by definition, $Q(t_1,t_2)\leq 1$ with probability one. By the discussion in the end of Section IV-B, we have

Corollary 4. *When $M=2$, the IDNC scheme achieves the capacity upper bound (1) when the file size tends to infinity.*

Remark 1: Note that the proofs of Corollary 4 and the propositions in Section IV-B are based only on the marginal distribution of $C_1(t)$ and $C_2(t)$, not on the joint channel distribution. Namely, when $M=2$, IDNC scheme is asymptotically optimal even when channels of the two users are spatially dependent (as long as they are memoryless in time).

Remark 2: Recall that for $M=2$, there is only one possible choice of non-empty coding group, but one still has the freedom of choosing which packets to be mixed together within the same coding group. The proof of Corollary 4 holds for any generic IDNC scheme, regardless of which packets are mixed together within the same coding group (e.g., we may choose either the oldest or the youngest packet first).

D. The Three-User Case

For $M\geq 3$, the dynamics of the system are much more complicated and the assumption of Proposition 2 may not always hold. For example, knowing that the marginal success probability being p is not sufficient to guarantee the optimal throughput of the IDNC schemes. More explicitly, suppose the channels are i.i.d. in time but spatially dependent such that only one user can successfully receive the packet at any time, e.g., $\mathsf{P}(C_1(t)=c_1,C_2(t)=c_2,C_3(t)=c_3)$ is equal to $\frac{1}{3}$ for $(c_1,c_2,c_3)\in\{(1,0,0),(0,1,0),(0,0,1)\}$, and is zero otherwise. With such a spatially dependent channel model, all packets will be accumulated in the lists L_{001}, L_{010}, and L_{100}. Since these three lists do not form a coding group (see Fig. 2), the BS is not able to send any IDNC packet and thus will transmit only non-coded packets. The corresponding normalized throughput is $\frac{\frac{1}{3}N}{3N}=1/9$, which is far from the upper bound $1/3$ in (1).

For the following, we show that when $M=3$, the IDNC scheme is asymptotically throughput optimal when applied to channels $(C_1(t),C_2(t),C_3(t))$ that are spatially independent and when we use the following priority policy. For ease of exposition, we first index all the coding groups as coding group 1 (L_{100} and L_{011}), coding group 2 (L_{010} and L_{101}), coding group 3 (L_{001} and L_{110}), and coding group 4 (L_{011}, L_{101}, and L_{110}). We call the lists L_{100}, L_{001}, and L_{010} as the lists in the "outer circle" (see Fig. 2), and L_{110}, L_{101} and L_{011} as the lists in the "inner circle".

The details of the priority policy is as follows: (1) Suppose the BS has transmitted an uncoded packet in time slot $t-1$, then at time slot t the BS checks in the order of coding groups 1, 2, 3, and 4 at time t. (2) Suppose the BS has transmitted an IDNC packet in $t-1$, then at time slot t the BS selects the coding group by the following rule: (a) If all coding groups are empty, then the BS sends an uncoded packet. (b) Else if the coding group m chosen in $t-1$ satisfies $1\leq m\leq 3$, then the BS checks in the order of coding group m, $(m+1) \mod 3$, $(m+2) \mod 3$, and 4. That is, the coding groups 1, 2, and 3 are checked in a counterclockwise order, as the arrows shown in Fig. 2, starting from the group that was used previously. Coding group 4 has the lowest priority. (c) Else if it is coding group 4 that was chosen in $t-1$, one can show that in time t, only coding group 4 can be non-empty. The BS simply continues to choose coding group 4 in t as well. (3) Finally if the BS chooses the coding group 1, 2, or 3, it selects the youngest packet from the list on the outer circle, and the oldest packet from the list on the inner circle; if the BS chooses the coding group 4, it selects the oldest packet from each list of coding group 4.

The main idea of the proof for $M=3$ is as follows. For $t_2>t_1$, let $|L_{\mathbf{v}}(t_1,t_2)|$ be the number of packets from $L_{\mathbf{v}}$ at the end of time t_2 when we only count those packets with indices $n\geq n(t_1)$. These packets are termed the "$L_{\mathbf{v}}$ packets in the current time-scope." The first step is to show that when $q(t_1)$ is large, for any $\tilde{B}<B$, if both $|L_{001}(t_1,t_1+\tilde{B})|$ and $|L_{100}(t_1,t_1+\tilde{B})|$ are large, then both $|L_{001}(t_1,t_1+\tilde{B})|$ and $|L_{100}(t_1,t_1+\tilde{B})|$ are of negative drift. Therefore, with high probability, at least one of $|L_{001}(t_1,t_1+\tilde{B})|$ and $|L_{100}(t_1,t_1+$

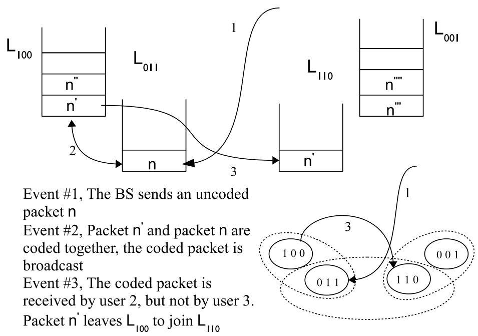

Fig. 4: Illustration of the chain effect on L_{100} and L_{001} (a)

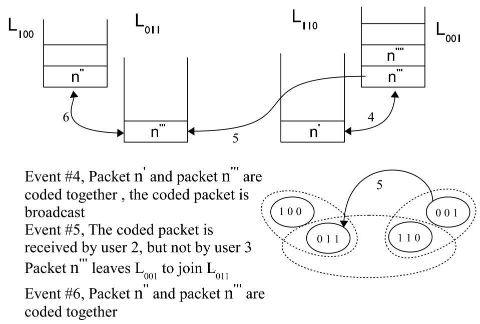

Fig. 5: Illustration of the chain effect on L_{100} and L_{001} (b)

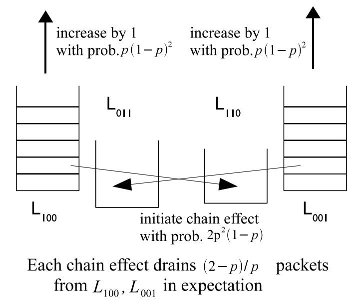

Fig. 6: High-level description of the chain effect

$\tilde{B})|$ remains finite/stable.

To show the negative drift, suppose that $|L_{001}(t_1, t_1 + \tilde{B})|$ and $|L_{100}(t_1, t_1 + \tilde{B})|$ are both large, and the BS decides to send an uncoded packet n. The only case that $|L_{001}(t_1, t_1+\tilde{B})|$ may increase is when user 3 is the only user that receives the uncoded transmission successfully, which is of probability $p(1-p)^2$. By considering the symmetric case that $|L_{100}(t_1, t_1 + \tilde{B})|$ may increase, with probability $2p(1-p)^2$, the summation $|L_{001}(t_1, t_1 + \tilde{B})| + |L_{100}(t_1, t_1 + \tilde{B})|$ may increase. On the other hand, if users 2 and 3 are the only users receiving the uncoded transmission, then packet n will enter the list L_{011}. What is interesting is that packet n then initiates a "chain effect" of draining packets from the lists L_{001} and L_{100} (symmetrically, the chain effect can also be initiated if packet n joins L_{110}). To see this, we first note that the new packet n in L_{011} will now be combined with another packet n' in L_{100} that is in the current time-scope (see Fig. 4). If this coded packet is successfully received by user 1, which occurs with probability p, then packet n will leave the system forever. Otherwise, packet n remains in the list L_{011} and can be mixed with one L_{100} packet. On average packet n can stay in L_{011} for $1/p$ times before it leaves the system. On the other hand, if the coded packet is received by users 2 or 3 (with probability $1-(1-p)^2$), then n' will leave the L_{100} list. Since packet n stays in L_{011} for $1/p$ time on average, in that duration on average $\frac{1-(1-p)^2}{p}$ packets will leave L_{100}. Moreover, some of those packets, say packet n', may enter the L_{110} list, if the coded packet is heard by user 2 but not by user 3. What is interesting is that packet n' (that moves into L_{110}) will then serve a symmetric role as that of original packet n in L_{011}, and trigger new packets moving from L_{001} to L_{011} (see Fig. 5). Those packets entering L_{011} will continue draining packets from L_{100} and start further chains of coded transmissions.

To analyze the net impact of the above "chain effect," we first assume that both $|L_{100}(t_1, t_1 + \tilde{B})|$ and $|L_{001}(t_1, t_1 + \tilde{B})|$ are sufficiently large. Let X denote on average the total number of packets in the current time scope that are drained from L_{001} and L_{100} due to a single packet n that enters into L_{011} at the beginning of the chain effect. Then, we have

$$X = \frac{1}{p}((1-(1-p)^2) + p(1-p)X) \Rightarrow X = \frac{2-p}{p}, \quad (5)$$

where $\frac{1}{p}$ is the average time that packet n stays in L_{011}; $1-(1-p)^2$ is the probability that in every coded transmission with packet n, one packet leaves L_{100}; $p(1-p)$ is the probability that, the coded packet is received by user 2 and not by user 3, in which case a L_{100} packet will leave L_{100} and join L_{110}. Note that those packet entering L_{110} will create new chain effects and the total depletion is thus described by (5). Since with probability $2p^2(1-p)$ a new packet n will enter either L_{110} or L_{011}, on average the depletion rate of L_{100} and L_{011} is $2Xp^2(1-p)$. Note that the probability that a new packet entering either L_{100} or L_{001} is $2p(1-p)^2$ (see Fig. 6). One can easily check by (5) that the depletion rate $2Xp^2(1-p)$ is strictly larger than the replenishing rate $2p(1-p)^2$. The overall drift for L_{001} and L_{100} is thus negative.

To rigorously carry out the above intuitive steps, we need to rely on the explicit priority policies of how to choose the non-empty coding group and how to choose the coding packets within the same group (as briefly described at the beginning of this subsection). Due to space constraints, we refer the readers to our technical report [22] for the detailed proofs and we only present the result of the analysis in the following lemma.

Lemma 5. *For spatially independent channels and with the appropriate priority policy, for any* $\epsilon_1' > 0$, *there exists* $B_1 > 0$, *such that for all* $B > B_1$ *and* $t_1 > 0$, *we have*

$$\begin{aligned} &\mathsf{P}\Big(|L_{001}(t_1, t_1 + B)| \geq \frac{\epsilon_1' \lambda B}{2} \text{ and} \\ &|L_{010}(t_1, t_1 + B)| \geq \frac{\epsilon_1' \lambda B}{2} \Big| q(t_1) > B\Big) < \frac{\epsilon_1'}{3}. \end{aligned} \quad (6)$$

Since (6) is true for any two lists on the outer circle, by the union bound we have

$$\begin{aligned} &\mathsf{P}\Big(\text{at least two } \mathbf{v}\text{'s in } \{100, 010, 001\} \text{ have} \\ &|L_{\mathbf{v}}(t_1, t_1 + B)| \geq \frac{\epsilon_1' \lambda B}{2} \Big| q(t_1) > B\Big) \leq \epsilon_1'. \end{aligned} \quad (7)$$

Next we explain the intuition how to use (7) to show that $Q(t_1, t_1 + B)$ satisfies (2). To that end, we first observe an intuitive principle, which is made rigorous in [22], that for any coding group, at least one of its constituent lists must be small. Otherwise, the lists in the coding group will be constantly mixed together, which reduces the sizes of all the lists.

According to (7), at most one out of the three lists in the outer circle can have a large number of packets in the current time scope. We then consider the following two cases. Case 1: Exactly one list has a large size, say $|L_{100}(t_1, t_1 + B)|$ is large. By definition, $|L_{100}| > |L_{100}(t_1, t_1 + B)|$ is also large. The intuitive principle then implies that $|L_{011}|$ should be small, which in turn implies that $|L_{011}(t_1, t_1+B)|$ is even smaller. By definition, $\mathsf{RC}_1(t_1, t_1+B) = |L_{010}(t_1, t_1+B)|+|L_{001}(t_1, t_1+B)|+|L_{011}(t_1, t_1+B)|$ is a summation of three small lists and is thus also small. Case 2: All three lists in the outer circle have a small number of packets in the current time scope. Since all lists in the inner circle belong to the same coding group, by the earlier intuitive principle, at least one of them, say list L_{101}, must be small, which in turn implies that $|L_{101}(t_1, t_1 + B)|$ is even smaller. By definition, $\mathsf{RC}_2(t_1, t_1 + B) = |L_{100}(t_1, t_1 + B)| + |L_{001}(t_1, t_1 + B)| + |L_{101}(t_1, t_1 + B)|$ is a summation of three small lists and is thus also small. The discussions of Cases 1 and 2 ensure that $Q(t_1, t_1 + B) = \min_j \mathsf{RC}_j(t_1, t_1 + B)$ is small with high probability. By formalizing the above arguments, we have

Proposition 6. *Consider an IDNC scheme with $M = 3$ and the policy rule as described in the beginning of Section IV-D. For any $\epsilon_1' > 0$, there exists $B_1 > 0$, such that for all $B > B_1$ and $t_1 > 0$,* $\mathsf{P}(Q(t_1, t_1 + B) \geq \epsilon_1' \lambda B | q(t_1) > B) \leq \epsilon_1'$.

Now by Proposition 2, we obtain the asymptotic throughput optimality for $M = 3$. The detailed proofs for Lemma 5 and Proposition 6 are available at [22]. Propositions 2, 3, and 6 jointly imply that the IDNC scheme with the specified priority policy is asymptotically optimal.

Remark: As readers have seen, Propositions 2 and 3 provide a more tractable sufficient condition for the asymptotic optimality of IDNC schemes, so that future work on asymptotically-optimal IDNC schemes can focus on designing the corresponding priority policy that satisfies (2).

V. Simulation

In this section, we use simulation to verify the performance of the IDNC scheme for finite N, and compare it with the uncoded case. For all of our simulation results, we assume that the deadline of the n-th packet is $d_n = 3n$, i.e., $\lambda = 3$.

One of our goals is to quantify the gains of comparing our IDNC scheme over the uncoded schemes. It is easy to see that some simple non-coding schemes will not perform as well as IDNC schemes. For example, suppose $M = 2$, $\lambda = 3$, and $p = 1/3$. The BS broadcasts each packet uncodedly for 3 consecutive time slots right before its deadline. The throughput of such a simple uncoded scheme can be analytically computed, and is equal to $\frac{N_{\text{success}}}{2N} = \mathsf{P}(\text{at least one copy is received by } d_1 \text{ in 3 time slots}) = 1 - (1 - p)^3 = 19/27$. The throughput of this simple non-coding scheme is strictly less than that of the IDNC scheme, which is $\lambda p = 1$.

To quantify the throughput improvement of IDNC schemes over the *best possible* uncoded scheme, we derive the optimal uncoded transmission scheme based on dynamic programming (DP) [23]. The closed-form expression of the expected throughput of the optimal DP policy is a complicated function of M and N. For fair comparison to the asymptotic throughput of IDNC schemes, we also derive in [22] the closed-form expression of the optimal uncoded scheme as a function of M when the file size N tends to infinity. For example, the asymptotic throughput for the best possible uncoded scheme with $M = 2$ is given by

$$\lim_{N\to\infty} \frac{\mathsf{E}\{N_{\text{success}}\}}{2N} = \begin{cases} 1 & \text{if } \frac{\lambda+1-\sqrt{\lambda^2+1-\lambda}}{\lambda} < p \leq 1 \\ \frac{1}{2}(\lambda p + \frac{1}{2-p}) & \text{if } \frac{\lambda-\sqrt{\lambda^2-\lambda}}{\lambda} < p \leq \frac{\lambda+1-\sqrt{\lambda^2+1-\lambda}}{\lambda} \\ \lambda p & \text{if } 0 < p \leq \frac{\lambda-\sqrt{\lambda^2-\lambda}}{\lambda} \end{cases}. \quad (8)$$

This asymptotic throughput is also an upper bound of any uncoded scheme with finite N. The detailed derivation of this upper bound and asymptotic throughput can be found in [22]. In our simulation results, we plot both the asymptotic throughput of the optimal uncoded scheme for general M values, and the optimality results in Sections IV-C and IV-D so that we can quantify the throughput improvement of IDNC schemes over the optimal uncoded schemes. In Fig. 7, we consider a large $N = 10000$ and plot the throughput of the IDNC scheme when compared to uncoded transmission for the cases of M =2, 3, 4, and 5, respectively. As illustrated in Fig. 7, the asymptotically optimal IDNC scheme indeed achieves the optimal throughput when $N = 10000$. As the number of users increases, the performance of uncoded transmission degrades, while the performance of the IDNC scheme achieves the broadcast channel capacity. In Figs. 8 and 9 we compare the IDNC policy of small, finite file size N for the two-user and three-user cases, respectively. Even for file size as small as $N = 50$ ($N = 100$), the performance of the IDNC scheme is better than the asymptotic throughput of the optimal uncoded scheme.

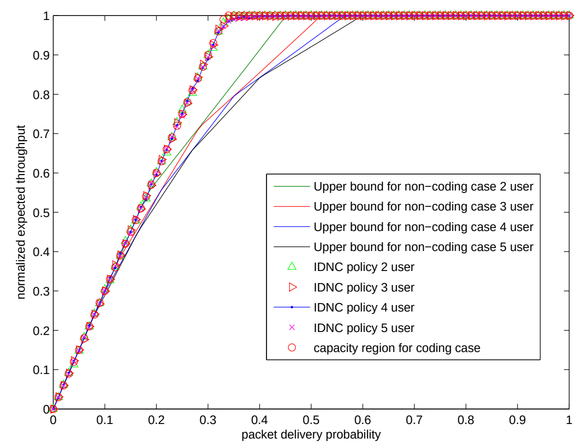

Fig. 7: The upper bound of best non-coding policy given in [22] and the actual throughput for the IDNC scheme.

We are also interested in the per-packet performance. Namely, if we focus only on the i-th packet, we count

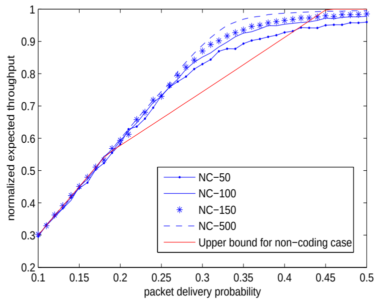

Fig. 8: Performance comparison between IDNC policies for small file size $N = 50$, 100, 150, 500, and the upper bound of uncoded case given in (8) (Two-user case).

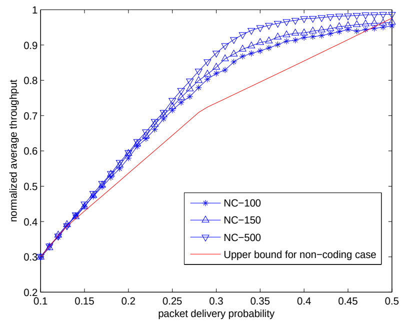

Fig. 9: Performance comparison between IDNC policies for small file size $N = 100$, 150, 500, and the upper bound of uncoded case given in [22] (Three-user case).

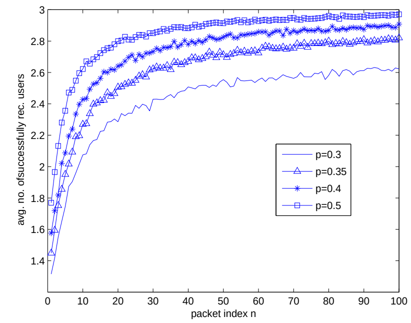

Fig. 10: The packet-by-packet delivery rate of the IDNC scheme for the first 100 packets (Three-user case). The deadline is $3n$ for the n-th packet.

how many users have received the i-th packet averaged over simulation runs. In Fig. 10, we plot the per-packet throughput for the case of $M = 3$ for the first 100 packets. We observe that the IDNC scheme achieves large throughput even in the initial phase. For those $p > \frac{1}{\lambda} = \frac{1}{3}$, the per-packet throughput approaches $M = 3$ as predicted by our analysis.

A. Extensions to The Settings of Imperfect Feedback

Although our theoretical results require instant & perfect feedback, we believe that IDNC schemes can also achieve good performance with imperfect feedback. For the following, we use simulation to study IDNC scheme for the practical setting in which the feedback is sent by each users once every t_{delay} time slots, and the feedback packet may get lost.

To account for infrequent and lossy feedback, we allow each feedback to contain a bit map of size t_{accumu} that informs the BS the reception status in the time interval $(t - t_{\text{accumu}}, t]$. This accumulative feedback provides sufficient redundancy, so that with high probability the BS can eventually receive the correct feedback for each packet transmitted. For a packet of receiving status vector $\mathbf{v}$, we define $U_{\text{ack}}(\mathbf{v})$ as the set of users who have not received/decoded that packet. For example, $U_{\text{ack}}(010) = \{1, 3\}$. Recall that the feedback is sent non-instantly and may be lost. For any packet n that is not in any list $L_{\mathbf{v}}$, we say packet n is *properly acknowledged* if either one of the following conditions is satisfied: (i) packet n has never been sent uncodedly before, or (ii) the BS has successfully received the feedback from all M users since the last time n was transmitted uncodedly. For any packet n that is in a list $L_{\mathbf{v}}$ for some $\mathbf{v}$, we say packet n is *properly acknowledged* if either one of the following conditions is satisfied: (i) packet n has never been involved in any coded transmission, or (ii) the BS has successfully received the feedback from all users in $U_{\text{ack}}(\mathbf{v})$ since the last time n was involved in a coded transmission. The IDNC scheme is described as follows.

1: Set $n \leftarrow 1$, set all $L_{\mathbf{v}} \leftarrow \emptyset$, for all $\mathbf{v} \neq (0 \dots 0)$ and $\mathbf{v} \neq (1 \dots 1)$. Let $\mathcal{I}_{\text{pkt}} \leftarrow \{1, \dots, N\}$ contain all packets that have not been received by any user.
2: **for** $t = 1$ to λN **do**
3: In the beginning of the t-th time slot, do the following:
4: In the following Lines 5 to 13, we consider only *properly acknowledged* packets. Namely, those packets that are not properly acknowledged are temporarily "suspended" and do not participate in any transmission.
5: **if** $\mathcal{I}_{\text{pkt}}$ is not empty **then**
6: **if** there exists at least one non-empty coding group **then**
7: Choose one non-empty coding group; generate and broadcast an IDNC coded packet.
8: **else**
9: Send the oldest packet in $\mathcal{I}_{\text{pkt}}$ uncodedly.
10: **end if**
11: **else**
12: Choose the oldest packet i in all $L_{\mathbf{v}}$
13: **end if**
14: In the end of the t-th time slot,
15: Remove all expired packets from all $L_{\mathbf{v}}$, $\mathcal{I}_{\text{pkt}}$.
16: **if** $(t \mod t_{\text{delay}}) = 0$ **then**
17: Each user sends accumulative feedback for reception status between $(t - t_{\text{accumu}}, t]$.
18: UPDATE all $L_{\mathbf{v}}$ and $\mathcal{I}_{\text{pkt}}$ based on the successful feedbacks received from the users.
19: **end if**
20: **end for**

The subroutine "UPDATE all $L_{\mathbf{v}}$ and $\mathcal{I}_{\text{pkt}}$ based on feedbacks" is described as follows: The BS only process those packets that were not properly acknowledged but become properly acknowledged in this time slot due to the arrival of new feedback packets. The rest of the UPDATE rules is identical to the perfect feedback setting in Section III-A.

The main idea behind this algorithm is that with imperfect feedback, the BS only processes the properly acknowledged packets, patiently waits for the arrival of new, cumulative feedback to "properly acknowledge the previously transmitted packets" and then updates the reception status accordingly.

To close this section, we evaluate the performance of our IDNC schemes under the above practical feedback setting for

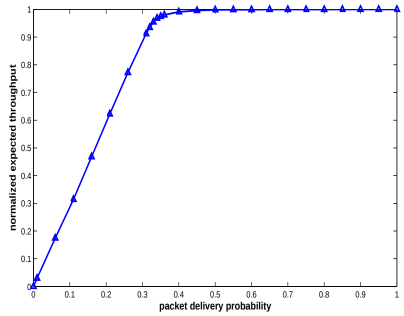

Fig. 11: Performance of IDNC policies with lossy and delayed feedback for file size $N = 20000$, with 8 users.

the 8-user case (Fig. 11). We assume that each user only sends feedback for every $t_{\text{delay}} = 3$ time slots, $t_{\text{accumu}} = 400$, and each feedback successfully arrives the BS with probability 0.9. As shown in Fig. 11, our IDNC scheme is robust and approaches the optimal throughput under the imperfect feedback setting and for a moderate number of users.

VI. Conclusions

In this work, we have modeled and analyzed the streaming broadcast problem over the downlink in a single cell for stored-video. We have proposed and analyzed a class of immediately decodable network coding (IDNC) transmission schemes, which asymptotically achieves the optimal throughput subject to deadline constraints without prior knowledge of the packet delivery probability. Compared with the generation-based scheme [6], the IDNC scheme achieves good throughput performance even in the initial period of transmission. By comparing the coded and uncoded cases, we analytically quantify the coding gain of a deadline-constrained system in the asymptotic sense. In addition, our simulation shows that the IDNC scheme achieves strictly higher throughput than that of the best uncoded scheme even for very small file size. A Lyapunov analysis of the *index advancement* has been developed, which sheds further insight into the dynamics of the IDNC schemes.

There are many interesting directions for future work. First, in this paper we have focused on symmetric channels. In such a symmetric setting, when the overall throughput is maximized, all users also receive fair service. An interesting question is whether IDNC scheme can still achieve asymptotically optimal throughput and maintain fairness in an asymmetric setting. Second, our definition of throughput treats all packets equally. Real video streams may react differently to each packet loss, depending on their importance in the video frame. Similarly, a long burst of losses may cause a different type of interruption than frequent but short bursts of losses. It would be interesting to see how we can generalize our formulation to take into account the different impact of each packet loss. Third, we have not paid much attention to the complexity of IDNC schemes (e.g., in searching for coding opportunities). Although our simulation indicates that the complexity is reasonable for up to 8 users, future work may study the low-complexity schemes for an even larger number of users.

Appendix A
Proof for Lemma 1

Proof: Define $\mathsf{US}_j(t_1, t_2)$ (which stands for "Uncoded Success") as the number of time slots in $(t_1, t_2]$ when user j receives an *uncoded* packet successfully; and define $\mathsf{UF}_j(t_1, t_2)$ (which stands for "Uncoded Failure") as the number of slots in $(t_1, t_2]$ when an *uncoded* packet is sent, user j fails to receive it, but some other users receive it successfully. Since the n variable increases when (and only when) an uncoded packet is sent and it is received by at least one user (see Line 15 of the pseudo-code), for any given user j we must have

$$n(t_2) - n(t_1) = \mathsf{US}_j(t_1, t_2) + \mathsf{UF}_j(t_1, t_2). \tag{9}$$

Note that the uncoded packet received by some other user but not by j creates a coding opportunity involving user j. By construction, all these coding opportunities have index $\geq n(t_1)$. Further, these coding opportunities remain unexpired in the end of time t_2 since $t_2 < \lambda n(t_1)$.

Define $\mathsf{CS}_j(t_1, t_2)$ (which stands for "Coded Success") as the number of time slots when user j receives a *coded* packet successfully during the interval $(t_1, t_2]$. We then notice that in each time slot when user j receives a coded packet successfully, user j can "immediately decode" that packet, which destroys a coding opportunity involving user j. Hence,

$$\mathsf{CS}_j(t_1, t_2) + \mathsf{RC}_j(t_1, t_2) \geq \mathsf{UF}_j(t_1, t_2). \tag{10}$$

The left-hand side of (10) is the number of coding opportunities destroyed due to successful decoding plus the number of remaining unexpired coding opportunities. The right-hand side of (10) is the number of coding opportunities created during the $(t_1, t_2]$ period. Since these $\mathsf{UF}_j(t_1, t_2)$ opportunities must either be destroyed during the $(t_1, t_2]$ period or being counted in $\mathsf{RC}_j(t_1, t_2)$ at the end of time t_2, we thus have (10).

By definition, $\mathsf{US}_j(t_1, t_2) + \mathsf{CS}_j(t_1, t_2) = C_j(t_1, t_2)$. Combining (9) and (10), we thus have

$$\begin{aligned} n(t_2) - n(t_1) &\leq \mathsf{US}_j(t_1, t_2) + \mathsf{CS}_j(t_1, t_2) + \mathsf{RC}_j(t_1, t_2) \\ &= C_j(t_1, t_2) + \mathsf{RC}_j(t_1, t_2), \text{ for all } j. \end{aligned}$$

Let $j^\star$ be the user with the smallest $\mathsf{RC}_j(t_1, t_2)$, i.e., $\mathsf{RC}_{j^\star}(t_1, t_2) = Q(t_1, t_2)$. We then have,

$$\begin{aligned} n(t_2) - n(t_1) &\leq C_{j^\star}(t_1, t_2) + Q(t_1, t_2) \\ &\leq \max_j C_j(t_1, t_2) + Q(t_1, t_2). \end{aligned}$$

■

Appendix B
Proof for Proposition 2

Proof: Consider the case $p < \frac{1}{\lambda}$. We will show that when $q(t) = B$ is sufficiently large, $\mathbb{E}\{q(t + \lambda B) - q(t) | q(t) = B\} < 0$. Namely, $q(t)$ has a negative drift when $q(t)$ is large. Note that when $q(t_1) = n(t_1) - \frac{t_1}{\lambda} > B$, we can find a pair $t_2 = t_1 + \lambda B$ and t_1 such that $t_2 < \lambda n(t_1)$. By Lemma 1,

$$\begin{aligned} &\mathsf{E}\{n(t_2) - n(t_1) | q(t_1) > B\} \\ &\leq \mathsf{E}\{\max_j C_j(t_1, t_2) + Q(t_1, t_2) | q(t_1) > B\}. \end{aligned}$$

Recall that channel conditions for any period $(t_1, t_2]$ are independent from random variable $q(t_1)$ at time t_1. For large enough $B = \frac{t_2 - t_1}{\lambda}$, by law of large numbers, we have

$$\begin{aligned} &\mathsf{E}\left\{ \max_j C_j(t_1, t_2) \middle| q(t_1) > B \right\} \\ &= \mathsf{E}\left\{ \max_j C_j(t_1, t_2) \right\} = p\lambda B + o(B). \end{aligned}$$

Since the assumption (2) indicates that $Q(t_1, t_2)$ cannot grow large if $q(t_1)$ is large, we have by Lemma 1, when B is sufficiently large

$$\mathsf{E}\{n(t_2) - n(t_1) | q(t_1) > B\} = p\lambda B + o(B).$$

By the definition of $q(t)$, we have

$$\begin{aligned} &\mathsf{E}\left\{ q(t_1 + \lambda B) - q(t_1) \right| q(t_1) > B\} \\ &= \mathsf{E}\left\{ \left(n(t_2) - \frac{t_2}{\lambda} \right) - \left(n(t_1) - \frac{t_1}{\lambda} \right) \middle| q(t_1) > B \right\} \\ &= p\lambda B + o(B) - \frac{\lambda B}{\lambda} < 0. \end{aligned} \tag{11}$$

Let $\epsilon_2 = \frac{1}{2}(1 - p\lambda)$. We thus have

$$\mathsf{E}\{q(t_1 + \lambda B) - q(t_1) | q(t_1) > B\} < -\epsilon_2 B.$$

■

References

[1] "Purdue estadium," *http://estadium.purdue.edu/*.

[2] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[3] S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[4] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, Nov. 2003.

[5] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[6] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. of Allerton Conference*, 2003.

[7] E. Drinea, C. Fragouli, and L. Keller, "Delay with network coding and feedback," in *Proc. of ISIT*, 2009.

[8] J.-K. Sundararajan, D. Shah, and M. Médard, "ARQ for network coding," in *Proc. of ISIT*, 2008.

[9] E. Erez, M. Effros, and T. Ho, "Network codes with deadlines," in *Proc. of Allerton Conference*, 2008.

[10] A. Eryilmaz, A. Ozdaglar, and M. Médard, "On delay performance gains from network coding," in *Proc. of CISS*, 2006.

[11] J.-K. Sundararajan, D. Shah, and M. Médard, "Online network coding for optimal throughput and delay – the three-receiver case," in *arXiv:0806.4264*, 2008.

[12] J.-K. Sundararajan, P. Sadeghi, and M. Médard, "A feedback-based adaptive broadcast coding scheme for reducing in-order delivery delay," in *NetCod*, 2009.

[13] W. Yeow, A. Hoang, and C. Tham, "Minimizing delay for multicast-streaming in wireless networks with network coding," in *Proc. of INFOCOM*, 2009.

[14] H. Gangammanavar and A. Eryilmaz, "Optimal coding and scheduling techniques for broadcasting deadline constraint traffic over unreliable wireless channels," in *Proc. of ISIT*, 2010.

[15] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *Proc. of ACM SIGCOMM*, 2006.

[16] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," in *Proc. of ACM MobiCom*, 2007.

[17] D. Traskov, M. Médard, P. Sadeghi, and R. Koetter, "Joint scheduling and instantaneously decodable network coding," in *Proc. of GLOBECOM Workshops*, 2009.

[18] J. Barros, R. Costa, D. Munaretto, and J. Widmer, "Effective delay control in online network coding," in *Proc. of INFOCOM*, 2009.

[19] L. Jiang and J. Walrand, "Stable and utility-maximizing scheduling for stochastic processing networks," in *Proc. of Allerton Conference*, 2009.

[20] D. Nguyen, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," in *NetCod*, 2007.

[21] X. Li, C.-C. Wang, and X. Lin, "Throughput and delay analysis on uncoded and coded wireless broadcast with hard deadline constraints," in *Proc. of INFOCOM, mini conference*, 2010.

[22] ——, "On the capacity of immediately-decodable coding schemes for wireless stored-video broadcast with hard deadline constraints," *Technical Report, Purdue University, http://docs.lib.purdue.edu/ecetr/406/*.

[23] ——, "Throughput and delay analysis on uncoded and coded wireless broadcast with hard deadline constraints," *Technical Report, Purdue University, http://docs.lib.purdue.edu/ecetr/398/*.

Xiaohang Li received the B.E. degree in Electronic Engineering and Information Science from University of Science and Technology of China, Hefei, China, in 2007. He is currently working towards the Ph.D. degree in Electrical and Computer Engineering at Purdue University, West Lafayette, IN. His research interests are in network coding, wireless network scheduling.

Chih-Chun Wang is currently an Assistant Professor of the School of Electrical and Computer Engineering of Purdue University. He received the B.E. degree in E.E. from National Taiwan University, Taipei, Taiwan in 1999, the M.S. degree in E.E., the Ph.D. degree in E.E. from Princeton University in 2002 and 2005, respectively. He worked in Comtrend Corporation, Taipei, Taiwan, as a design engineer in 2000 and spent the summer of 2004 with Flarion Technologies, New Jersey. In 2005, he held a post-doctoral researcher position in the Department of Electrical Engineering of Princeton University. He joined Purdue University as an Assistant Professor in 2006. His current research interests are in the graph-theoretic and algorithmic analysis of iterative decoding and of network coding. Other research interests of his fall in the general areas of networking, optimal control, information theory, detection theory, and coding theory. Dr. Wang received the National Science Foundation Faculty Early Career Development (CAREER) Award in 2009.

Xiaojun Lin (S'02 / M'05) received his B.S. from Zhongshan University, Guangzhou, China, in 1994, and his M.S. and Ph.D. degrees from Purdue University, West Lafayette, Indiana, in 2000 and 2005, respectively. He is currently an Assistant Professor of Electrical and Computer Engineering at Purdue University.

Dr. Lin's research interests are resource allocation, optimization, network pricing, routing, congestion control, network as a large system, cross-layer design in wireless networks, mobile ad hoc and sensor networks. He received the IEEE INFOCOM 2008 best paper award and 2005 best paper of the year award from *Journal of Communications and Networks*. His paper was also one of two runner-up papers for the best-paper award at IEEE INFOCOM 2005. He received the NSF CAREER award in 2007. He has served as the Workshop co-chair for IEEE GLOBECOM 2007, the Panel co-chair for WICON 2008, and the TPC co-chair for ACM MobiHoc 2009.