12-21-2010

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

Xiaohang Li

Purdue University - Main Campus, li179@purdue.edu

Chih-Chun Wang

Purdue University - Main Campus, chihw@purdue.edu

Xiaojun Lin

Electrical and Computer Engineering, Purdue University, linx@purdue.edu

Follow this and additional works at: <http://docs.lib.purdue.edu/ecetr>

Li, Xiaohang; Wang, Chih-Chun; and Lin, Xiaojun, "On The Capacity of Immediately-Decodable Coding Schemes for Wireless Stored-video Broadcast with Hard Deadline Constraints" (2010). *ECE Technical Reports*. Paper 406.
<http://docs.lib.purdue.edu/ecetr/406>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

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

Xiaohang Li

Chih-Chun Wang

Xiaojun Lin

TR-ECE-10-11

December 21, 2010

School of Electrical and Computer Engineering

1285 Electrical Engineering Building

Purdue University

West Lafayette, IN 47907-1285

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
 School of ECE, Purdue University, West Lafayette, IN 47907
 Email: {li179, chihw, linx}@purdue.edu

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 innovative¹ to all receivers rather than only a subset of the receivers [7]. 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 types² 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, [8] discusses how different methods of encoding can affect the decoding delay while considering only noise-free channels. [9] 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. [7],

Xiaohang Li, Chih-Chun Wang, and Xiaojun Lin are with the Department of Electrical and Computer Engineering, Purdue University, Indiana, IN, 47907 USA e-mail: {li179, chihw, linx}@purdue.edu).

¹A 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], [13].

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

[10], [11], [12] 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. [13] 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. However, 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.

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 d_1 and d_2 are interested in different packets X and Y , respectively, and also suppose that d_1 has overheard Y and d_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 d_1 and d_2 ’s perspectives. Compared to generation-based solutions, IDNC schemes (i) have substantially smaller decoding delay, (ii) require much smaller buffers at the sender 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 analytically quantify 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.

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 VI provides some numerical comparison of IDNC and non-coding schemes. Then we conclude. Detailed proofs are relegated to the appendices.

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 $n = 1, 2, \dots, N$ 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 n 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 $j = 1, 2, \dots, M$, 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 $(C_1(t), \dots, C_M(t))$ is i.i.d. for all t , the marginal success probability is $P(C_j(t) = 1) = p$ 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 [13], [7], [10], [11], [18], [20], [21].

Fig. 1. Illustration of various delays

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
$RC_j(t_1, t_2)$	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 $\min_j RC_j(t_1, t_2)$
$C_j(t_1, t_2)$	Number of channel successes for user j between time $(t_1, t_2]$
L_v	The list of packets with receiving status v

Fig. 3. Some key notations used in this work

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 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 be “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 $\frac{\mathbb{E}\{N_{\text{success}}\}}{MN}$.

A. An Upper Bound on the Optimal Throughput

To derive an upper bound of the 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, $\mathbb{E}\{N_{\text{success}}\} \leq \mathbb{E}\left\{\sum_{t=1}^{\lambda N} \sum_{j=1}^M C_j(t)\right\} = M\lambda Np$. Further, since the best scenario is that each user can recover/decode all N information packets, we also have $\mathbb{E}\{N_{\text{success}}\} \leq MN$. Jointly, we have

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

We next show that for $M \leq 3$, we can design 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 $(D_1(n) D_2(n) \dots D_M(n))$. At the end of the t -th slot, we define L_v as the list of unexpired packets such that the receiving status $\text{Rec}(n)$ is v , where v is an M -dimensional binary vector that is neither all-zero nor all-one.³ More explicitly, $L_v \triangleq \{n : \lambda n > t, \text{Rec}(n) = v\}$. 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 $(\text{Rec}(n_1)^T, \text{Rec}(n_2)^T, \dots, \text{Rec}(n_K)^T)$. If for every row of this matrix, the integer sum of all elements in this row is exactly $K - 1$, we call this matrix an immediately decodable coding-opportunity matrix, and these K packets can be added together, 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

³We do not consider $L_{11\dots 1}$ and $L_{00\dots 0}$ since $L_{11\dots 1}$ is the list of packets that have been received by all users, while $L_{00\dots 0}$ is the list of packets that have not been received by any users.

who successfully receives the coded packet can *immediately decode* the remaining packet that it has not received before. In this work, we simply use “coding-opportunity matrix” as shorthand for immediately decodable coding-opportunity matrix. If packets from different lists can form an IDNC packet, then these lists are said to constitute a coding group. A coding group is indexed by the corresponding coding opportunity matrix. 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. If a coding group consists of i lists, then we call it a type- i 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.

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.

```

1: Set  $n \leftarrow 1$ , set all  $L_v \leftarrow \emptyset$ , for all  $v \neq (0 \dots 0)$  and  $v \neq (1 \dots 1)$ .
2: for  $t = 1$  to  $\lambda N$  do
3:   In the beginning of the  $t$ -th time slot, do the following:
4:   if  $n \leq N$  then
5:     if there exists at least one non-empty coding group then
6:       Choose one non-empty coding group; generate and broadcast an IDNC coded packet .
7:     else
8:       Send uncoded packet  $n$  directly.
9:     end if
10:  else
11:    Choose the oldest4 packet  $i$  in all  $L_v$ , and send packet  $i$  uncodedly.
12:  end if
13:  In the end of the  $t$ -th time slot,
14:  if we sent an uncoded packet in time  $t$ , and it is received by at least one user then
15:     $n \leftarrow n + 1$ .
16:  end if
17:  UPDATE all  $L_v$ , based on the feedback received from all users.
18: end for

```

The subroutine “UPDATE all L_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 v to v' , then we remove packet n from L_v and add it to $L_{v'}$. If it is an uncoded packet that is transmitted and suppose its receiving status is $\text{Rec}(n) = v$ that is neither all-zero nor all-one, then we add packet n to the list L_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 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 .

⁴The 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: $q(t) \triangleq n(t) - \frac{t}{\lambda}$. Note that if $q(t)$ remains finite with high probability when $N \rightarrow \infty$, 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 $N \rightarrow \infty$ (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.

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_v will have the j -th bit of receiving status v being 0. Namely, the packet in that list L_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 coding opportunities involving user 2 is $|L_{100}| + |L_{101}| + |L_{110}|$. 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 time slots $t_1 < t_2$, if $t_2 < \lambda n(t_1)$, then*

$$n(t_2) - n(t_1) \leq \max_j C_j(t_1, t_2) + Q(t_1, t_2),$$

where $Q(t_1, t_2) = \min_j RC_j(t_1, t_2)$, and $RC_j(t_1, t_2)$ 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 $(t_1, t_2]$.)

Proof: Define $US_j(t_1, t_2)$ (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 $UF_j(t_1, t_2)$ (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

$$n(t_2) - n(t_1) = US_j(t_1, t_2) + UF_j(t_1, t_2). \quad (2)$$

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 $CS_j(t_1, t_2)$ (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,

$$CS_j(t_1, t_2) + RC_j(t_1, t_2) \geq UF_j(t_1, t_2). \quad (3)$$

The left-hand side of (3) is the number of coding opportunities destroyed due to successful decoding plus the number of remaining unexpired coding opportunities. The right-hand side of (3) is the number of coding opportunities created during the $(t_1, t_2]$ period. Since these $UF_j(t_1, t_2)$ opportunities must either be destroyed during the $(t_1, t_2]$ period or being counted in $RC_j(t_1, t_2)$ at the end of time t_2 , we thus have (3).

By definition, $US_j(t_1, t_2) + CS_j(t_1, t_2) = C_j(t_1, t_2)$. Combining (2) and (3), we thus have

$$\begin{aligned} n(t_2) - n(t_1) &\leq US_j(t_1, t_2) + CS_j(t_1, t_2) + RC_j(t_1, t_2) \\ &= C_j(t_1, t_2) + RC_j(t_1, t_2), \text{ for all } j. \end{aligned}$$

Let j^* be the user with the smallest $RC_j(t_1, t_2)$, i.e., $RC_{j^*}(t_1, t_2) = Q(t_1, t_2)$. We then have,

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

■

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 (4), then $q(t)$ has a negative drift whenever $q(t)$ is large.

Proposition 2. *Suppose $p < \frac{1}{\lambda}$, $\forall \epsilon'_1 > 0$, $\exists B_1 > 0$, such that for all $B > B_1$ and $t_1 > 0$*

$$P(Q(t_1, t_1 + B) \geq \epsilon'_1 \lambda B | q(t_1) > B) < \epsilon'_1, \quad (4)$$

then $q(t)$ has a negative drift when $q(t)$ is large.

Proof: Recall that the channel conditions for any period $(t_1, t_2]$ are independent from the random variable $q(t_1)$ at time

t_1 . By the law of large numbers, for any $\epsilon > 0$, and $\delta > 0$, we can find a B_2 such that for all $B \geq B_2$ and $t_2 = t_1 + B$,

$$\begin{aligned} & \mathbb{P} \left(\frac{\max_j C_j(t_1, t_2)}{t_2 - t_1} - p > \epsilon \mid q(t_1) > B \right) \\ &= \mathbb{P} \left(\frac{\max_j C_j(t_1, t_2)}{t_2 - t_1} - p > \epsilon \right) < \delta. \end{aligned} \quad (5)$$

Reuse the B_1 value in the assumption of this proposition and consider any $\bar{B} > \max(B_1, B_2)$. Suppose $q(t_1) = n(t_1) - \frac{t_1}{\lambda} > \bar{B}$. By definition, choosing $t_2 = t_1 + \lambda \bar{B}$ will satisfy $t_2 < \lambda n(t_1)$. By Lemma 1, the assumption of Proposition 2, (5), and the union bound, we then have

$$\mathbb{P} \left(n(t_2) - n(t_1) \leq (p + \epsilon + \epsilon'_1) \lambda \bar{B} \mid q(t_1) > \bar{B} \right) \geq 1 - \delta - \epsilon'_1. \quad (6)$$

Since the n variable increments by at most 1 in each time slot, we also have $n(t_2) - n(t_1) \leq t_2 - t_1 = \lambda \bar{B}$. Jointly we have

$$\begin{aligned} & \mathbb{E} \{ n(t_2) - n(t_1) \mid q(t_1) > \bar{B} \} \\ & \leq ((p + \epsilon + \epsilon'_1) \lambda \bar{B}) (1 - \delta - \epsilon'_1) + \lambda \bar{B} (\delta + \epsilon'_1). \end{aligned}$$

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

$$\begin{aligned} & \mathbb{E} \{ q(t_1 + \lambda \bar{B}) - q(t_1) \mid q(t_1) > \bar{B} \} \\ &= \mathbb{E} \left\{ \left(n(t_2) - \frac{t_2}{\lambda} \right) - \left(n(t_1) - \frac{t_1}{\lambda} \right) \mid q(t_1) > \bar{B} \right\} \\ & \leq \left(((p + \epsilon + \epsilon'_1)(1 - \delta - \epsilon'_1) + \delta + \epsilon'_1) - \frac{1}{\lambda} \right) \lambda \bar{B}. \end{aligned} \quad (7)$$

For any $p < \frac{1}{\lambda}$, one can choose sufficiently small ϵ , ϵ'_1 , δ , and then sufficiently large $\bar{B}$ such that the drift value in (7) is strictly negative. ■

Proposition 2 shows that if $p < 1/\lambda$, and condition (4) 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 $t_1 > 0$

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

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

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 $\mathbb{P}(q(t) < \epsilon' t) > 1 - \epsilon$, for $t > t_0$. By plugging in the definition of $q(t)$, we thus have

$$1 - \epsilon < \mathbb{P} \left(n(t) - t/\lambda < \epsilon' t \right) = \mathbb{P} \left(\frac{Mpt}{Mn(t)} > \frac{p\lambda}{1 + \lambda\epsilon'} \right). \quad (8)$$

for all $t > t_0$, where the equality of (8) 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 (8) implies that the normalized throughput satisfies

$$\lim_{t \rightarrow \infty} \mathbb{E} \left\{ \frac{Mpt}{M(n(t) - 1)} \right\} \geq \frac{p\lambda}{1 + \lambda\epsilon'} (1 - \epsilon). \quad (9)$$

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 $\lim_{N \rightarrow \infty} \frac{\mathbb{E} \{ N_{\text{success}} \}}{MN} \geq \lambda p$. By the upper bound in Section II-A, we also have that for the case of $p < 1/\lambda$, $\frac{\mathbb{E} \{ N_{\text{success}} \}}{MN} \leq \lambda p$. 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 (4). In the following, we will see that $Q(t_1, t_2)$ satisfies (4) trivially for the case of $M = 2$, and we will prove that $Q(t_1, t_2)$ satisfies (4) 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.

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 shown as follows.

Now we consider the case when $p > \frac{1}{\lambda}$. In this case, we choose a $\lambda' < \lambda$ such that $\bar{\epsilon} = \frac{1}{\lambda'} - p > 0$. Note that our choice of λ' ensures that $p < \frac{1}{\lambda'}$. We now define a new, auxiliary advancement function $q'(t) = n(t) - \frac{t}{\lambda'}$ and will show that $q'(t)$ has a negative drift when $q'(t) > B$.

By the law of large numbers, for any $\epsilon > 0$, and $\delta > 0$, we can find a B_2 such that for all $B \geq B_2$ and $t_2 = t_1 + B$,

$$\mathbb{P}\left(\frac{C_j(t_1, t_2)}{t_2 - t_1} - p > \epsilon | q'(t_1) > B\right) = \mathbb{P}\left(\frac{C_j(t_1, t_2)}{t_2 - t_1} - p > \epsilon\right) < \delta.$$

Reuse the B_1 value in the assumption of Proposition 2 and consider any $\bar{B} > \max(B_1, B_2)$. Suppose $q'(t_1) = n(t_1) - \frac{t_1}{\lambda'} > \bar{B}$. By definition, choosing $t_2 = t_1 + \lambda' \bar{B}$ will satisfy $t_2 < \lambda' n(t_1) < \lambda n(t_1)$. We can then following the same steps as used in deriving (7) and (8), we can show that $q'(t)$ has a negative drift conditioning on that $q'(t) > \bar{B}$.

By similar arguments as used in the case of $p < \frac{1}{\lambda}$, the negative drift of $q'(t)$ can then be used to show that the normalized throughput satisfies

$$\lim_{t \rightarrow \infty} \mathbb{E} \left\{ \frac{Mpt}{M(n(t) - 1)} \right\} \geq \frac{p\lambda'}{1 + \lambda'\epsilon' - \lambda'/t} (1 - \epsilon).$$

Since the above statement holds for any ϵ' satisfying $\frac{1}{\lambda'} > p$, we can choose a λ' such that $\lambda' \cdot p$ is arbitrarily close to one. This in turn implies that the normalized throughput satisfies

$$\lim_{t \rightarrow \infty} \mathbb{E} \left\{ \frac{Mpt}{M(n(t) - 1)} \right\} \geq 1$$

The proof for the $p \geq \frac{1}{\lambda}$ case is thus complete.

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: We note that the proofs of Corollary 4 and the key 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 $M = 2$, an IDNC scheme is asymptotically optimal even when the 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., $\mathbb{P}(C_1(t) = c_1, C_2(t) = c_2, C_3(t) = c_3)$ 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).

Moreover, for the case of $M = 2$, there exists only one coding group and thus each time there is an opportunity to form an IDNC packet, it must correspond to the only coding group and we only need to decide which packets in this coding group the BS should mix together. However, for $M \geq 3$ there might be multiple non-empty coding groups in any time slot. It is thus essential that we decide among all non-empty coding groups, which one has the higher priority so that the BS should choose it first. *It turns out that under certain channel models, the performance of an immediately decodable coding scheme could depend on the choice of which coding group to encode first, in a similar way as the impact of scheduling policies to any delay sensitive applications.* More explicitly, when there are $M = 3$ users, the coding groups can be indexed by 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}). As shown in Fig. 2, the lists within one dotted-line-circle belong to one coding group. Note that only coding group 4 is

	$\text{Rec}(n_1)^T$	$\text{Rec}(n_2)^T$
user 1	1	0
user 2	1	0
user 3	0	1

	$\text{Rec}(n_1)^T$	$\text{Rec}(n_2)^T$	$\text{Rec}(n_3)^T$
user 1	1	0	1
user 2	0	1	1
user 3	1	1	0

Fig. 4. Illustration of coding-opportunity matrix for coding group 3 and 4 for the three-user case

of type 3, and others are of type 2.⁵ For easy reference, 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”. 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 among different coding groups:

- Suppose in the previous time slot, the BS has chosen one coding group, then in the current time slot the incumbent coding group always has the highest priority.
- Among all the non-incumbent coding groups, a coding group of a smaller type is of higher priority than a coding group. Therefore, coding groups 1, 2, and 3 are given higher priority than coding group 4.
- More specifically, if in the previous time slot, the BS has transmitted a non-coding packet. Then the priority is given to coding groups 1, 2, 3, and 4 in this order. Line 6 thus searches whether the coding group is non-empty according to the give order. If in the previous time slot, the BS has transmitted a group- m coding packet for some $m \in \{1, 2, 3\}$, then the priority is given to the cyclic permutation $(m + 1), \dots, 3, 1, \dots, m - 1$ of the first three groups, and then given to coding group 4. If in the previous time slot, the BS has transmitted a group-4 coding packet, then the priority is given to coding groups 4, 1, 2, and 3 in this order.
- Once a non-empty coding group is chosen according to the above priority rule, for any list of the outer circle, choose the *youngest* packet to mix. For any list of the inner circle, choose with the *oldest* packet to mix. The intuition behind this “youngest in the outer circle, oldest in the inner circle” policy will be provided shortly after.

We would also like to emphasize that the above specifics provide provable convergence guarantees. Other priority policies, such as always selecting the oldest packet to mix, seem to work well in our simulation.

Next we will present the idea for proving the throughput optimality of the three-user case. Recall that the BS only transmits a coded or an uncoded packet in one slot. To that end, we introduce the concept of the “super time slot”: the n -th super slot starts from the beginning of the time slot in which the n -th uncoded packet is transmitted for the first time, and ends before the beginning of the time slot when the $n + 1$ -th uncoded packet is transmitted for the first time. Namely, the n -th super slot starts when the BS cannot form any IDNC packets and the BS has to send the n -th packet for the very first time. Sending an uncoded packet n may also create new opportunities to form IDNC packets. During the n -th super time slot, the system will focus on exploiting any opportunities to form IDNC packets created by the uncoded transmission of packet n . Once all such opportunities are finished (which may be due to successful reception or due to packet expiration), the system moves on to the $(n + 1)$ -th super time slot.

In the following, we consider the interval $(t_1, t_2]$ with $t_2 = t_1 + B$. For simplicity, we define $\mathcal{N}_1 \triangleq n(t_1)$. Let $L_{\mathbf{v}}(t_1, t_2)$ denote the set of packets from $L_{\mathbf{v}}$ at the end of time t_2 when we only count those packets with indices $\geq \mathcal{N}_1$. For $t_2 > t_1$, let $|L_{\mathbf{v}}(t_1, t_2)|$ be the number of packets in $L_{\mathbf{v}}(t_1, t_2)$. For $n \geq \mathcal{N}_1$, let $L_{\mathbf{v}}(t_1, t_{\text{end}}(n))$ denote the set of packets in $L_{\mathbf{v}}$ at the end of super slot n (that is, the end of time slot $t_{\text{end}}(n)$) when we only count those packets with indices $\geq \mathcal{N}_1$. Since in this proof, we focus only on the packets with indices $\geq \mathcal{N}_1$, we term those packets as the “ $L_{\mathbf{v}}$ packets in the current time-scope $(t_1, t_2]$ ”. For simplicity, we also use $L_{\mathbf{v}}(t_1, n)$ as shorthand of $L_{\mathbf{v}}(t_1, t_{\text{end}}(n))$ when it is clear from the context that we are focusing on the end of the n -th super time slot.

Next we are going to show that when $q(t_1)$ is large, with high probability only one of $L_{001}(t_1, n)$, $L_{010}(t_1, n)$, and $L_{100}(t_1, n)$ can be large. Define $p_{\mathbf{v}}$ as the probability that when an uncoded packet is transmitted, this packet enters the list $L_{\mathbf{v}}$. For example, suppose the BS transmits a packet n , and only user 1 receives it, then it enters L_{100} , and this probability is $p_{100} = p(1 - p)^2$.

The first step is to show that when $q(t_1)$ is large, for $n \geq \mathcal{N}_1$, if both $|L_{001}(t_1, n - 1)|$ and $|L_{100}(t_1, n - 1)|$ are large, then both $|L_{001}(t_1, n - 1)|$ and $|L_{100}(t_1, n - 1)|$ are of negative drift. Therefore, with high probability, at least one of $|L_{001}(t_1, n)|$ and $|L_{100}(t_1, n)|$ remains finite/stable. To show the negative drift, suppose that $|L_{001}(t_1, n - 1)|$ and $|L_{100}(t_1, n - 1)|$ are both large, and the BS decides to send an uncoded packet n . Since at the beginning of any super slot, no packets can be coded together, we also assume that L_{011} and L_{110} are empty at the beginning of super slot n (obviously $L_{011}(t_1, n - 1)$ and $L_{110}(t_1, n - 1)$ are thus empty). Note that since our priority policy selects youngest packet from the lists on the outer circle, so when we refer to the coded transmission which involve packets from the outer circle, we only consider those packets with indices $\geq \mathcal{N}_1$. The only case that $|L_{001}(t_1, n)|$ 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 $|L_{100}(t_1, n)|$ may increase, with probability $2p(1 - p)^2$, the summation $|L_{001}(t_1, n)| + |L_{100}(t_1, n)|$ 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

⁵We also give an illustration of the coding-opportunity matrix for coding groups 3 and 4 in Fig. 4.

Fig. 5. Illustration of the chain effect on L_{100} and L_{001} (a) Fig. 6. Illustration of the chain effect on L_{100} and L_{001} (b)

n then initiates a “chain effect” of draining packets from the lists L_{001} and L_{100} if we use the *youngest-packet-first* policy. To see this, we first note that the new packet n will now be combined with another packet n' in L_{100} that is in the current time-scope. 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 stay in L_{011} for $1/p$ time, in that duration $\frac{1-(1-p)^2}{p}$ 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} . Those packets entering L_{011} will continue draining packets from L_{100} and start further chains of coded transmissions. See Fig. 5 and 6 for illustration. To analyze the net impact of the above “chain effect,” we first assume that both $|L_{100}(t_1, n - 1)|$ and $|L_{001}(t_1, n - 1)|$ are sufficiently large. Let X denote 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

$$X = \frac{1}{p}((1 - (1 - p)^2) + p(1 - p)X) \Rightarrow X = \frac{2 - p}{p}, \quad (10)$$

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, for which 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 (10). 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

Fig. 7. High-level description of the chain effect

either L_{100} or L_{001} is $2p(1-p)^2$. One can easily check by (10) 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. By carefully analyzing the chain effect, we have the following proposition:

Proposition 5. *For spatially independent channels and with an 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*

$$\mathbb{P}\left(\text{more than two } \mathbf{v}'\text{'s in } \{100, 010, 001\} \text{ have } |L_{\mathbf{v}}(t_1, t_1 + B)| \geq \frac{\epsilon'_1 \lambda B}{2} \mid q(t_1) > B\right) \leq \epsilon'_1. \quad (11)$$

The detailed proof is available in Appendix A.

Next we explain the intuition how to use (11) to show that $Q(t_1, t_1 + B)$ must be bounded by $\epsilon'_1 \lambda B$ with high probability conditioning on $q(t_1) > B$. To that end, we first observe an intuitive principle, which is made rigorous in the proof of Proposition 6, 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 (11), 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 cases. Case 1: suppose exactly one such list, 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, $\text{RC}_1(t_1, t_1 + B) = |L_{010}(t_1, t_1 + B)| + |L_{001}(t_1, t_1 + B)| + |L_{011}(t_1, t_1 + B)|$ is a summation of three small lists and is thus also small. Case 2: suppose 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, $\text{RC}_2(t_1, t_1 + B) = |L_{100}(t_1, t_1 + B)| + |L_{001}(t_1, t_1 + B)| + |L_{101}(t_1, t_1 + B)|$ is a summation of three small lists and is thus also small. The discussions of Cases 1 and 2 ensure that $Q(t_1, t_1 + B) = \min_j \text{RC}_j(t_1, t_1 + B)$ is small with high probability. By formalizing the above arguments, we have

Proposition 6. *For $M = 3$, for any $\epsilon'_1 > 0$, there exists $B_1 > 0$, such that for all $B > B_1$ and $t_1 > 0$, $\mathbb{P}(Q(t_1, t_1 + B) \geq \epsilon'_1 \lambda B \mid q(t_1) > B) \leq \epsilon'_1$.*

The detailed proof is relegated to Appendix B. Proposition 6, and together with the discussion of Section IV-B prove the asymptotical throughput optimality of IDNC schemes for the case of 3 users.

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 (4).

V. THE OPTIMAL THROUGHPUT FOR UNCODED TRANSMISSION

For the uncoded case, the optimal transmission policy can be solved by first formulating the problem as a Markov decision problem, and then devising the optimal policy via dynamic programming techniques. Such an optimal dynamic programming policy was explicitly formulated for the two-user case in the [22], and it can be easily extended to the multiple-user case considered herein. Although the dynamic programming policy can be efficiently computed numerically, a closed-form expression of the optimal policy is hard to obtain. In the following subsections, we instead derive a closed-form, asymptotically tight upper bound on the optimal throughput for the uncoded transmissions, which characterizes the performance of the optimal

w_0	+1	+1	+1			+1			+1
w_1					+1		+1		
w_2				+1				+1	
time t	1	2	3	4	5	6	7	8	9
packet n	1	2	3	2	1	4	4	1	5
User1	rec.	rec.	rec.	rec.	E	rec.	E	E	E
User2	E	rec.	rec.	E	rec.	E	rec.	rec.	rec.

Fig. 8. Illustration of w_i . “E” means erasure, and “rec.” means successfully received by a user

dynamic programming policy when N is asymptotically large. In the numerical experiments of Section VI, we use this upper bound for the uncoded transmission as a benchmark, which can be used to quantify the throughput improvement of the IDNC schemes.

A. An Upper Bound for the Optimal Throughput

We first obtain an upper bound for the optimal throughput in the uncoded case for any give N by relaxing the deadline constraints. Namely, we quantify the maximum achievable performance when all packets have the same common deadline λN (instead of individual deadlines λn for packet n). When comparing the setting of this subsection with that of the capacity outer bound in Section II-A, we now limit ourselves to consider only uncoded transmission policy while Section II-A considers both coded and uncoded policies. To that end, we first categorize the packet broadcast at a given time t into M types: types- $0, 1, \dots, M$. We say a packet of type- m is transmitted in time t if there are exactly m users who have received this packet (to-be-transmitted in time t) in the previous time slots $[1, t-1]$.

Consider any transmission policy. Let $w_0, \dots, w_M$ denote the numbers of time slots that are used to transmit packets of type- $0, 1, \dots, M$, respectively. Note that $w_0, \dots, w_M$ are random variables depending on the underlying channel realizations and on the uncoded transmission policy of interest.

Fig. 8 illustrates the construction of w_0 to w_2 for a given policy and channel realization when there are two users in the system. Under the given policy and channel realization, we can compute the value of w_0 to w_2 as follows. Before all the transmissions, all w_j are set to be 0. In Fig. 8, at the beginning of time slot 1, packet 1 has not been received by any user before. So packet 1 at time 1 is classified as a packet of type-0. Since the BS schedules a type-0 packet for this time slot, w_0 is increased by 1. Similarly, packets 2 and 3 scheduled at times 2 and 3 are also of type-0, which contributes to the increment of w_0 at times 2 and 3. At the beginning of time slot 4 the BS decides to retransmit packet 2 according to the underlying policy. (Here we allow any arbitrary policies including both optimal and suboptimal ones.) Since packet 2 has been received by both users at the end of $t=2$, at time slot 4 packet 2 is classified as type-2. Therefore w_2 increases by 1. For $t=5$, packet 1 has been received by 1 user already and is thus classified as type-1. Therefore, w_1 is increased by 1. After all 9 transmissions, we have that $w_0=5$, $w_1=2$ and $w_2=2$. Note that w_0 , w_1 and w_2 are random variables depending on the channel realization and the underlying policy.

Let $\bar{w}_0, \dots, \bar{w}_M$ denote the expectation of $w_0, \dots, w_M$, respectively. When the BS transmits a packet of type-0 to all users, the expected reward in the time slot is exactly Mp . When the BS transmits a packet of type- j , $0 < j < M$, the expected reward in the time slot is $(M-j)p$. Transmitting a packet of type- M would not contribute to the throughput, so for any reasonable uncoded transmission policy, we can assume $w_M=0$. Using the optional sampling theorem for Martingales, we can show that the expected total number of successes is $\sum_{j=0}^M (M-j)p\bar{w}_j$.

Further, each packet of type-0 (that is transmitted when it has not been received by any users), with probability $Mp(1-p)^{M-1}$ it will be received by exactly one user, which creates a packet of type-1. When transmitting a packet of type-1, with probability $1-(1-p)^{M-1}$ it will be received by one or more users, which destroys a packet of type-1 (while creating a packet of type-2, 3, $\dots, M$). Again by the optional sampling theorem for Martingales and by the conservation law of type-1 packets, we have

$$\bar{w}_1 (1 - (1-p)^{M-1}) \leq \bar{w}_0 Mp(1-p)^{M-1}. \quad (12)$$

By noting that $\sum_{j=0}^M w_j = \lambda N$ and by the conservation law of all types packets in a similar way as in (12), we have

$$\max \mathbb{E}\{N_{\text{success}}\} = \sum_{j=0}^M (M-j)p\bar{w}_j \quad (13)$$

$$\text{s.t.} \quad \sum_{j=0}^M \bar{w}_j \leq \lambda N \quad (14)$$

$$\bar{w}_0(1 - (1-p)^M) \leq N \quad (15)$$

$$\bar{w}_j(1 - (1-p)^{M-j}) \leq \sum_{s=0}^{j-1} \bar{w}_s \binom{M-s}{j-s} p^{j-s} (1-p)^{M-j}, \quad j = 1, \dots, M-1 \quad (16)$$

$$\bar{w}_j \geq 0 \quad j = 0, \dots, M. \quad (17)$$

Since the above constraints hold for any policy, we can thus upper bound the best achievable rate for uncoded transmission by maximizing (13) subject to the constraints in (14)-(17). A closed-form solution to this linear program then produces an upper bound for the original problem of maximizing throughput with hard deadline constraints for each packet. We call this problem as Upper Bound Linear Program (UBLP) for the uncoded case.

B. An Asymptotic Tight Lower Bound for the Optimal Throughput When $N \rightarrow \infty$

Next we will show that the upper bound given by (13) subject to the constraints in (14)-(17) can be achieved when $N \rightarrow \infty$. To this end, we will construct an even simpler policy that attains a matching lower bound on the optimal throughput.

Suppose we temporarily mark the time at which the BS decides to transmit packet n for the first time as the new origin, which will be used to define $X_n^1, X_n^2, \dots$, and X_n^M . Let X_n^1 denote the number of additional time slots it takes before the packet broadcast channel successfully carry one more packet from the BS to user 1. For example, suppose in the beginning of time 7, the BS for the very first time, decides to transmit the 5-th packet. If user 1 does not receive the packet transmitted at time 7 but receives the packet transmitted at time 8, then $X_5^1 = 2 = 8 - (7 - 1)$. Note that the packet that is actually passed from the BS to user 1 may not be the n -th packet. The n -th packet is only used to mark the beginning of the X_n^1 consecutive time slots. Since the channel is i.i.d. with delivery probability p , we have $P(X_n^1 = i) = (1-p)^{i-1}p$ for all n . We define $X_n^j, j = 2, \dots, M$ by symmetry. Given variables $X_n^1, X_n^2, \dots, X_n^M$, the order statistics $X_n^{(1)}, X_n^{(2)}, \dots, X_n^{(M)}$ are also random variables, defined by sorting the values (realizations) of $X_n^1, X_n^2, \dots, X_n^M$ in increasing order. Actually, $X_n^{(1)}$ denotes the number of time slots it takes before at least one user has received at least one packet. Again, here the n -th packet is only used to mark the beginning of the $X_n^{(1)}$ consecutive time slots. $X_n^{(2)}, \dots, X_n^{(M)}$, denotes the number of time slots before 2, $\dots, M$ users receive at least one packet, respectively.

Consider the following transmission policies. In the first policy, for any ongoing packet (say packet n), repeatedly transmit it until all users receive it or the packet expires. Then move to packet $n+1$, repeatedly transmit packet $n+1$, until all M users receive it or the packet expires, and so on. We denote this policy by π_M . The policy $\pi_k, k = 1, 2, \dots, M-1$, is similar to policy π_M , and the difference is that the BS keeps transmitting the ongoing packet until the packet is received by at least k users, respectively. Then the BS moves on to the next packet.

We now consider several schemes that perform *random mixtures of* $\pi_1, \dots, \pi_M$, and dropping packets. In our mixed policy, for the very first time that the BS would like to transmit the n -th packet, it has $M+1$ options: transmit a packet based on policy π_k for $k = 1, \dots, M$, respectively, or the BS can simply drop the current packet and move to the next packet. (For notational simplicity, we can denote this action of dropping a packet as policy π_0). We use $P(\pi_k)$ to denote the probability that the BS picks π_k , for $k = 0, 1, \dots, M$. Need to note that $\sum_{k=0}^M P(\pi_k) = 1$. The BS chooses *randomly and independently* among these options. Once the sub-policy is decided, the BS uses the chosen policy to transmit packet n . For the next packet $n+1$, the BS will again choose randomly among these sub-policies. We then have the following lemma for the above mixed policy.

Lemma 7. If $\sum_{k=1}^M P(\pi_k) \mathbb{E}\{X_n^{(k)}\} \leq \lambda$, then

$$\lim_{N \rightarrow \infty} \frac{\mathbb{E}\{N_{\text{success}}\}}{N} = \sum_{k=1}^M P(\pi_k) \left(k + \sum_{l=k+1}^M P(X_n^{(l)} = X_n^{(k)}) \right).$$

The detailed proof for Lemma 7 can be found in Appendix C. Lemma 7 describes the expected throughput when we randomly and independently choose the $\pi_1, \dots, \pi_M$ policies. Now we are going to show that for the uncoded case, how to select the “optimal” mixture probability $P(\pi_k)$, $0 \leq k \leq M$, to achieve the the largest possible expected throughput. It turns out that

finding the optimal mixture probability can be formulated as the following linear programming problem.

$$\max \sum_{k=1}^M P(\pi_k) \left(k + \sum_{l=k+1}^M P(X^{(l)} = X^{(k)}) \right) \quad (18)$$

$$\text{s.t.} \quad \sum_{k=1}^M P(\pi_k) E\{X^{(k)}\} = \lambda - \epsilon_1 \quad (19)$$

$$\sum_{k=1}^M P(\pi_k) \leq 1 \quad (20)$$

$$P(\pi_k) \geq 0, \quad k = 0, 1, \dots, M., \quad (21)$$

where $X^{(l)}$ is shorthand for the order statistics $X_n^{(l)}$ for which we drop the subscript since the distribution of $X_n^{(l)}$ is identical for any n value. By solving (18) subject to (19)-(21) with a sufficiently small ϵ_1 , the constructed mixed policy thus achieves the lower bound for the optimal normalized expected throughput. We call this problem as the Lower Bound Linear Program (LBLP) for the uncoded case.

C. The Upper Bound Is Asymptotically Tight

In the following, we will show that the lower bound and the upper bound are actually equal when $\epsilon_1 \rightarrow 0$ and when $N \rightarrow \infty$. In particular, given (13) subject to (14)-(17), we can construct a solution for (18) subject to (19)-(21), such that the objective value of (18) equals to (13) when $\epsilon_1 \rightarrow 0$. As the first step of this goal, we study the special structure of the solution to the UBLP for the uncoded case.

Lemma 8. *The solution to (13) subject to (14)-(17) depends on whether λN is less than $\frac{N}{1-(1-p)^M}$ or not. More explicitly, the optimal solution has the following forms:*

Case 1: when $\lambda N < \frac{N}{1-(1-p)^M}$, the optimal solution is $\bar{w}_0 = \lambda N$, $\bar{w}_j = 0, 1 \leq j \leq M$.

Case 2: when $\lambda N \geq \frac{N}{1-(1-p)^M}$, there exists a $m \in (1, M-1]$ such that the optimal solution is $\bar{w}_0 = \frac{N}{1-(1-p)^M}$,

$$\begin{aligned} \bar{w}_j(1 - (1-p)^{M-j}) &= \sum_{s=0}^{j-1} \bar{w}_s \binom{M-s}{j-s} p^{j-s} (1-p)^{M-j} \\ &\text{for } 1 \leq j < m, \\ \bar{w}_j(1 - (1-p)^{M-j}) &\leq \sum_{s=0}^{j-1} \bar{w}_s \binom{M-s}{j-s} p^{j-s} (1-p)^{M-j} \\ &\text{for } j = m, \\ \bar{w}_j &= 0, \quad \text{for } m < j \leq M. \end{aligned} \quad (22)$$

After proving Lemma 8 in Appendix D, we use the optimal $\bar{w}_j$, $j = 0, \dots, M-1$, to construct $P(\pi_k)$, $k = 1, \dots, M$, by solving the following equality,

$$\bar{w}_0 = E\{X^{(1)}\} N \sum_{k=1}^M P(\pi_k) \quad (23)$$

$$\bar{w}_j = E\{X^{(j+1)} - X^{(j)}\} N \sum_{k=j+1}^M P(\pi_k), \quad j = 1, 2, \dots, M-1, \quad (24)$$

where $X^{(l)}$ is shorthand for the order statistics $X_n^{(l)}$ for which we drop the subscript since the distribution of $X_n^{(l)}$ is identical for any n value. Intuitively, $NP(\pi_k)E\{X^{(1)}\}$, represents the expected total number of time slots that contribute to $\bar{w}_0$, when the BS chooses to use the π_k policy, $k = 1, 2, \dots, M$, respectively. Similarly, for each j , $NP(\pi_k)E\{X^{(j+1)} - X^{(j)}\}$, $k = j+1, \dots, M$, represents the expected total number of time slots that contribute to $\bar{w}_j$, when the BS chooses to use the π_k policy, $k = j+1, \dots, M$, respectively.

For ease of notation, let $\bar{w}$ denote $(\bar{w}_0, \dots, \bar{w}_{M-1})^T$, and $P(\pi)$ denote $(P(\pi_1), \dots, P(\pi_M))^T$. Then (23) and (24) can be written as $\bar{w} = \Lambda P(\pi)$, where Λ is a matrix of coefficients. It's easy to verify that Λ is an upper-triangular matrix, and all the elements on the diagonal are not zero, which means we can obtain a unique $P(\pi)$ from $\bar{w}$ by solving $\bar{w} = \Lambda P(\pi)$.

We are interested in the relationships between the UBLP and the LBLP. Starting from $\bar{w}$, the optimal solution to (13) with the constraints of (14)-(17), we need to show that (i) the $P(\pi)$ constructed by (23) and (24) satisfy the inequality in (19)-(21), and (ii) the objective function value in (18) matches the value in (13).

By (15), we can easily show that (20) is satisfied once we plug in the newly constructed values of $P(\pi)$. More explicitly, since we already know $E\{X^{(1)}\} = \frac{1}{1-(1-p)^M}$, then equalities (23) and (15) jointly imply (20).

We can also obtain (19) from (14). Namely, We first note that $\sum_{j=0}^M \bar{w}_j \leq \lambda N$ by (14). If we plug the expressions of (23) and (24) into (14), then we can simplify $\sum_{j=0}^M \bar{w}_j \leq \lambda N$ as follows.

$$\sum_{k=1}^M P(\pi_k) E\{X^{(k)}\} N \leq \lambda N. \quad (25)$$

Then as ϵ_1 becomes arbitrarily small, we have (19).

Now we are going to show that the constructed $P(\pi_k) \geq 0$, for $1 \leq k \leq M$, which can be shown by Lemma 9 and Proposition 10 as follows.

Lemma 9. *Given the optimal $\bar{w}$ in (22), for $k > m + 1$, $P(\pi_k) = 0$.*

The proof for Lemma 9 is relegated to Appendix E.

Proposition 10. *Given the optimal solution to the UBLP in Lemma 8, the computed $P(\pi_k)$, $1 \leq k \leq M$ must be of the following forms.*

If $\lambda N < \frac{N}{1-(1-p)^M}$, then $P(\pi_1) = \lambda(1 - (1-p)^M)$, and $P(\pi_k) = 0$, $1 < k \leq M$.

If $\lambda N \geq \frac{N}{1-(1-p)^M}$, then

$$\begin{aligned} P(\pi_{m+1}) &= \frac{\bar{w}_m(1 - (1-p)^{M-m})}{\sum_{s=0}^{m-1} \bar{w}_s \binom{M-s}{m-s} p^{m-s} (1-p)^{M-m}} \leq 1 \\ P(\pi_m) &= 1 - \frac{\bar{w}_m(1 - (1-p)^{M-m})}{\sum_{s=0}^{m-1} \bar{w}_s \binom{M-s}{m-s} p^{m-s} (1-p)^{M-m}} \geq 0 \\ P(\pi_k) &= 0, \text{ if } k \neq m, \text{ or } k \neq m+1, \end{aligned}$$

where the m value is obtained from the optimal solution in Lemma 8. In short, once we compute $P(\pi)$ according to (23) and (15) there are at most two k values satisfying $P(\pi_k) > 0$.

The detailed proof for Proposition 10 is available in Appendix G. Therefore we can conclude that all constraints in (19)-(21) can be deduced from (14)-(17). That is, given optimal $\bar{w}$, the constructed $P(\pi)$, obtained by solving (23) and (24), satisfy all the constraints in (19)-(21). The final step is to show that the constructed $P(\pi)$ results in the same objective value in (18) as in (13).

Theorem 11. *If constructing $P(\pi)$ from $\bar{w}$ by solving (24) and (23), then we have*

$$\begin{aligned} & \frac{\sum_{j=0}^M (M-j)p\bar{w}_j}{N} \\ &= \sum_{k=1}^M P(\pi_k) \left(k + \sum_{l=k+1}^M P(X^{(l)} = X^{(k)}) \right). \end{aligned} \quad (26)$$

In summary, starting from the UBLP, we can deduce the solution to the LBLP, when $N \rightarrow \infty$. We thus conclude that the objective value of the solution to the UBLP is less than or equal to that of the LBLP. Recall that the solution to the UBLP is an upper bound to any non-coded policy, including the mixed policy that corresponds to the LBLP problem. We thus conclude that the upper bound matches with the lower bound for the optimal throughput of the uncoded transmission. The solution to (13) subject to the constraints (14)-(17) is indeed the optimal throughput of the non-coding case⁶.

VI. SIMULATION

Our previous analyses focus on the asymptotic case when $N \rightarrow \infty$. 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 our simulation results, we assume that the deadline of the n -th packet is $d_n = 3n$, i.e., $\lambda = 3$.

A. Performance for Large N

In Fig. 9, 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. 9, the asymptotic optimal IDNC scheme

⁶The expected normalized throughput can be obtained by $\frac{E_{\text{success}}}{MN}$

Fig. 9. The upper bound of non-coding policy and the actual throughput for the IDNC scheme.

Fig. 10. Performance comparison between IDNC policies for small file size $N = 50, 100, 150, 500$, and the upper bound of uncoded case (Two-user case).

indeed achieves the optimal throughput when $N = 10000$. As the number of users increases, the performance of uncoded transmission degrades, while the performance of IDNC scheme achieves the broadcast channel capacity. For the proposed IDNC policy, for each given p value, we run the simulation and count the number of successes for all users.

B. Performance for Small N

In Fig. 10 we compare the IDNC policy for small, finite file size N , with the upper bound of the uncoded transmission of two-user case. Even for file size as small as $N = 50$, the performance of IDNC scheme is better than the best of uncoded case. The expected throughput of the IDNC scheme deviates slightly more from its asymptotic expression for small N (i.e., $N < 500$). The performance degradation of the IDNC scheme at small N is due to the following reason. Initially, the *index advancement* $q(t)$ is small, which means that the ongoing packet n that have recently been transmitted are going to expire quickly (with the deadline λn close to t). Due to the randomness of the channel, those initial packets have a larger probability to expire, which affects the throughput. Further, we compare the IDNC policies for $N = 50, 150$, and 500 with the upper bound of uncoded transmission for three-user case (Fig 11). We can see that the performance of IDNC scheme when the file size is small can beat even the upper bound of all non-coding schemes. Here the performance of coded transmission is slightly worse than the two-user case, which can be explained as: when the BS has a chance to mix a coded packet which is innovative to two users, it probably has to wait a few more time slots so that it can send a packet that is innovative to all users. During this period, some packets may get expired, thus the throughput suffers a little bit.

In Fig. 12, for two-user case we perform multiple experiments and calculate the averaged total number of successes (for both users) for the first 50 packets $n = 1$ to 50. For example, for the case in which $p = 0.5$, among 2000 different realizations of the NC scheme, the first packet $n = 1$ has been successfully received/decoded in average by ≈ 1.4 users. All packets with index $n \geq 6$ have been successfully received/decoded on average by ≥ 1.8 users, which is above 90% of the achievable throughput.

Fig. 11. Performance comparison between IDNC policies for small file size $N = 100, 150, 500$, and the upper bound of uncoded case (Three-user case).

Fig. 12. The packet-by-packet delivery rate of the IDNC scheme for first 50 packets (Two-user case). The deadline is $3n$ for the n -th packet.

Even for a noisy environment $p = 0.35$, which is close to the critical delivery probability $p^* = \frac{1}{\lambda} = 1/3$, the first packet is received/decoded by ≈ 1.15 users, and 90% of the optimal throughput (avg. 1.8 users) can be achieved after $n \geq 26$. When $p = 0.3 < p^*$, the maximal achievable throughput is $p\lambda = 0.9$. The per-packet throughput for $p = 0.3$ is thus upper bounded by avg. 1.8 users as also illustrated in Fig. 12. The relatively large packet loss for the initial packets (those with small n) is the cause of the throughput degradation in Fig. 10. For example, for the case in which $p = 0.5$, the total area under the curve from $n = 1$ to $n = 50$ is approximately 96, which means that there are roughly 4% throughput losses in the first 50 packets. This 4% loss is also illustrated in Fig. 10 by the intersecting point of the “NC-50” curve and the $p = 0.5$ vertical line.

Similar results for three-user case is shown in Fig 13. We run the simulation for $N = 300$ case, and plot the averaged total number of successes (for all users) for the first 100 packets $n = 1$ to 100. For the case in which $p = 0.5$, the total area under the curve from $n = 1$ to $n = 100$ is approximately 285, which means that there are roughly 5% throughput losses in the first 50 packets. This 5% loss is also illustrated in Fig. 11 by the intersecting point of the “NC-100” curve and the $p = 0.5$ vertical line.

We also plot the average number of users that receive the n -th packet before the deadline for $n = 400$ to 450 in Fig 14 (two-user case). When $p = 0.35, 0.4$, and $p = 0.5$, all packets with indices between 400 and 450 are received by almost 2 users on average. This means nearly 100 % throughput can be achieved. This is because by this time, the index advancement $q(t) = n(t) - \frac{t}{\lambda}$ has grown to a sufficiently large value. The probability of deadline violation will be small. When $p = 0.3$ being less than the critical probability $\frac{1}{\lambda} = \frac{1}{3}$, the normalized capacity is 0.9 as proven in Section IV-A. As illustrated in Fig 14, the per-packet throughput for $p = 0.3$ approaches the upper bound $1.8 = 0.9 \times 2$ users for the packets with indices between 400 and 450. This also verifies the asymptotic optimality of the proposed scheme.

On the other hand, for two-user case when N is large, the initial loss of 4 packets in the first 50 packets is averaged over all N packets. Therefore, the asymptotic performance of large N approaches the broadcast capacity, as predicted in Section IV-A

Fig. 13. The packet-by-packet delivery rate of the IDNC scheme for first 100 packets (Three-user case). The deadline is $3n$ for the n -th packet.

Fig. 14. The packet-by-packet delivery rate of the NC scheme for packets with indices between 400 and 450. The deadline is $3n$ for the n -th packet

and verified in Fig. 9.

C. Time Evolution of $q(t)$ for two-user case

Fig. 15 shows the time evolution of the index advancement $q(t) = n(t) - \frac{t}{\lambda}$ for $p = 0.33$, which contains the trajectories of the $q(t)$ for 40 random realizations. As predicted in Section IV-A, $q(t)$ remains small (≤ 85) for the entire duration $t \in [1, 5000]$. Among 1000 random realizations, only 70 of the $q(t)$ curves have ever been over 85. As shown in (7), the smaller the p value is, the larger the negative drift is going to be. This phenomenon can also be verified in simulation. In another simulation (Fig. 16) with $p = 0.25$ we found that the index advancement $q(t)$ for all 40 realizations are upper bounded by 15.

In addition to its role in network coded throughput, the index advancement $q(t)$ is also highly relevant to *transmission delay* in the setting of *sequential packet arrival*. More explicitly, suppose that instead of transmitting a single file, we consider live video for which not all packets are available in the beginning of the broadcast session. In live video streaming, suppose the n -th packet arrives at the BS at time $\lambda n - \Delta$, where $\Delta > 0$ is the time offset between the arrival time at the BS and the deadline λn at the end users. This Δ thus represents the *maximum allowable transmission delay* that includes the queueing, propagation, and decoding delays. Note that in the proposed NC protocol, the packet sent at t_0 is generated (either codedly or non-codedly) by packets of index $\leq n(t_0)$. If the $n(t_0)$ -th packet has already arrived at the BS by time t_0 , i.e., if

$$\begin{aligned} t_0 &\geq \lambda n(t_0) - \Delta = \lambda(q(t_0) + \frac{t_0}{\lambda}) - \Delta \\ &\Leftrightarrow \Delta \geq \lambda q(t_0), \end{aligned}$$

then the NC protocol, originally proposed for file streaming with all packets available in the beginning of the session, can also be applied to the sequential-arrival live streaming applications with maximum transmission delay Δ . The analysis in Section IV-A

Fig. 15. Time evolution of the index advancement $q(t)$ for $p = 0.33$.

Fig. 16. Time evolution of the index advancement $q(t) = n(t) - \frac{t}{\lambda}$ for $p = 0.25$.

shows that, the NC scheme achieves close to optimal throughput for a sequential arrival setting with a sufficiently large Δ . The simulation results show that with $\lambda = 3$, $p = 0.33$ (resp. $p = 0.25$), if the maximum allowable delay is $\Delta = 85 \times 3$ (resp. $\Delta = 14 \times 3$), then in 93.0% (resp. 96.7%) of the 1000 realizations, the NC scheme can achieve the optimal throughput of live-video streaming under the maximum allowable delay constraint Δ .

D. 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. Our simulation results show that such an IDNC scheme is still asymptotically optimal.

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 then described as follows.

Fig. 17. Performance of IDNC policies with lossy and delayed feedback for file size $N = 20000$, with 8 users.

- 1: Set $n \leftarrow 1$, set all $L_v \leftarrow \emptyset$, 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_v
- 13: **end if**
- 14: In the end of the t -th time slot,
- 15: Remove all expired packets from all $L_v, \mathcal{I}_{\text{pkt}}$.
- 16: **if** $(t \bmod t_{\text{delay}}) = 0$ **then**
- 17: Each user sends accumulative feedback for reception status between $(t - t_{\text{accumu}}, t]$.
- 18: UPDATE all L_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_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 the 8-user case (Fig. 17). 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. 17, our IDNC schemes is robust and approaches the optimal throughput under the imperfect feedback setting and for a moderate number of users.

ACKNOWLEDGEMENT

This work was supported in part by NSF grants CCF-0845968, CNS-0643145, CNS-0721484, CNS-0813000, CNS-0905331.

VII. 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. Future work can continue to investigate the following aspects.

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 PROPOSITION 5

Proof: To simplify the analysis, we first assume that packets never expire.

We first assume $L_{001}(t_1, n-1)$ and $L_{100}(t_1, n-1)$ have an infinitely large number of packets, i.e., this also implies that L_{110} and L_{011} are empty at the end of time slot $t_{\text{end}}(n-1)$. Therefore, $L_{110}(t_1, n-1)$ and $L_{011}(t_1, n-1)$ are empty as well. Since we choose the youngest packet from the list on the outer circle, when some packets leave L_v (for a given $v \in \{001, 010, 100\}$), such a packet departure (that is the youngest among all packets in L_v) will decrease $|L_v(t_1, n)|$ unless $|L_v(t_1, n)| = 0$.

Now we discuss how $L_{100}(t_1, n)$ will change by considering the dynamics during the n -th super time slot. At the first time slot of super slot n , the BS transmits the n -th uncoded packet. This packet has probability $p_{100} = p(1-p)^2$ to enter L_{100} , for which the length of $L_{100}(t_1, n)$ increases by 1. Also with probability $p_{011} = p^2(1-p)$, the packet n will enter L_{011} . For notational simplicity, we use E_{011} to denote the event that the following conditions are satisfied: (i) $q(t_1) > B$, and (ii) the uncoded packet n enters L_{011} at the beginning of super slot n . We call this packet n “the 1st generation packet in L_{011} ”. In the next time slot, this 1st generation packet in L_{011} will be mixed with a packet from L_{100} . Such a coding operation is called the start of the “1st round transmission with L_{011} ”. Due to our policy of giving the incumbent coding group the highest priority, a sequence of totally $Y_1^1(1)$ packets will leave L_{100} until the coded transmission is received by user 1, which destroys packet n from the list L_{011} and thus the incumbent coding group (L_{100}, L_{011}) is empty. Note that $Y_1^1(1)$ is a random variable depending on the channel realization. When packet n , the 1st generation packet in L_{011} , leaves L_{011} , we say the 1st round transmission with L_{011} is finished. Among all $Y_1^1(1)$ packets leaving L_{100} , totally $Z_1^1(1)$ packets will actually enter L_{110} when the coded transmission is heard by user 2 but not by user 3. (Note that some packets will leave L_{100} to enter L_{101} if the coded transmission is heard by user 3 but not by user 2.) As will be clear shortly after, we use $Y_1^1 = Y_1^1(1)$ to denote the total number of packets that leave L_{100} , and $Z_1^1 \triangleq Z_1^1(1)$ to denote the total number of packets that leave L_{100} and enter L_{110} in the 1st round transmission with L_{011} . We also define $\tilde{Y}_1^1 = 0$, and $\tilde{Z}_1^1 = 0$.

After the end of the 1st round transmission with L_{011} , coding group 1 becomes empty. Our priority policy will then turn its focus to coding group 2. If both L_{101} and L_{010} are nonempty at this time, the “1st round transmission with L_{101} ” will begin. During the 1st round transmission with L_{101} , some packets will leave L_{010} and join L_{011} or join L_{110} .

After the end of the 1st round transmission with L_{101} , coding group 2 becomes empty and the priority policy turns its focus to coding group 3. The “1st round transmission with L_{110} ” begins. We first note that each of those Z_1^1 packets (that come from L_{100} during the 1st round transmission with L_{011}) will now be mixed with packet from L_{001} . Totally, there are $Y_1^2 \triangleq Y_1^2(1) + Y_1^2(2) + \dots + Y_1^2(Z_1^1)$ packets leaving L_{001} , where $Y_1^2(k)$ is the number of packets leaving L_{001} when mixing a L_{001} packet with the k -th packet (out of the Z_1^1 packets) of L_{110} . Among them, $Z_1^2 \triangleq Z_1^2(1) + Z_1^2(2) + \dots + Z_1^2(Z_1^1)$ packets will leave L_{001} and enter L_{011} , where $Z_1^2(k)$ is the number of such packets when mixing a L_{001} packet with the k -th packet (out of the Z_1^1 packets) of L_{110} . Moreover, besides those Z_1^1 packets (that come from L_{100}), other packets (mainly including those who leave L_{010} and join L_{110} during the 1st round transmission with L_{101}) will also be coded with packets in L_{001} . These coded transmission will cause $\tilde{Y}_1^2$ packets to leave L_{001} , and among them $\tilde{Z}_1^2$ packets will leave L_{001} to join L_{011} . The 1st round transmission with L_{110} ends when L_{110} is empty. Overall, in the “1st round transmission with L_{110} ”, $Y_1^2 + \tilde{Y}_1^2$ packets have left L_{001} , and among them, $Z_1^2 + \tilde{Z}_1^2$ have left L_{001} and enter L_{011} . Also note that during this period some packets may leave L_{001} to join L_{101} .

When the “1st round transmission with L_{110} ” ends, all the packets in L_{011} will be mixed with packets from L_{100} . We say that the “2nd round transmission with L_{011} ” starts. Each of these Z_1^2 packets will be coded with packets from L_{100} , which make $Y_2^1 \triangleq Y_2^1(1) + \dots + Y_2^1(Z_1^2)$ packets leave L_{100} , where $Y_2^1(k)$ is the number of packets leaving L_{100} when mixing a L_{100} packet with the k -th packet (out of the Z_1^2 packets) of L_{011} . Among these packets, $Z_2^1 \triangleq Z_2^1(1) + \dots + Z_2^1(Z_1^2)$ of them

will enter L_{110} . Similarly, all other packets that enter L_{011} in the previous rounds (including those who come from L_{010} in “1st round transmission with L_{101} ”, and those $\tilde{Z}_1^2$ packets that come from L_{001} in “1st round transmission with L_{110} ”) will also be coded with packets from L_{100} , and altogether make additional $\tilde{Y}_2^1$ packets leave L_{100} . Among them, $\tilde{Z}_2^1$ packets leave L_{100} to join L_{110} . Overall, in the 2nd round transmission with L_{011} , totally $Y_2^1 + \tilde{Y}_2^1$ packets leave L_{100} , and $Z_2^1 + \tilde{Z}_2^1$ packets leave L_{100} to join L_{110} . Also note that some packets will leave L_{100} and join L_{101} .

When the 2nd round transmission with L_{011} ends, the priority policy will turn to the coding group (L_{010}, L_{101}) , and the 2nd round transmission with L_{101} will begin. Before exhausting all opportunities to form IDNC packets within this coding group, some packets will leave L_{010} and join L_{011} and L_{110} , respectively. After the 2nd round transmission with L_{101} , the priority policy will go to the 2nd round transmission with L_{110} . During the 2nd round transmission with L_{110} , $Y_2^2 + \tilde{Y}_2^2$ packets will leave L_{001} , where Y_2^2 denotes those packets that leave L_{001} due to the coded transmission with those Z_2^1 packets that come from L_{100} , and $\tilde{Y}_2^2$ denotes the packets that leave L_{001} due to the coded transmission with packets other than those Z_2^1 packets that come from L_{100} . Among them, $Z_2^2 + \tilde{Z}_2^2$ packets will leave L_{001} , where Z_2^2 denotes those packets that leave L_{001} and join L_{011} during the coded transmission wose Z_2^1 packets that come from L_{100} , and $\tilde{Z}_2^2$ denotes those packets that leave L_{001} and join L_{011} among those $\tilde{Y}_2^2$ packets.

The above round-based notation can be defined iteratively. As the coded transmission proceeds round by round, more and more packets will leave L_{100} . During the coded transmission, if within any round r for any user, Z_r^1 or $Z_r^2 = 0$, we define all the subsequent $Z_i^1, Z_i^2 = 0$, for all $i > r$.

With the above iterative notation, when an uncoded packet enters L_{011} , the expected total number of packets that will leave L_{100} within R rounds of coded transmission with both L_{100} , L_{010} , and L_{001} is

$$\mathbb{E} \left\{ Y_1^1 + \tilde{Y}_1^1 + Y_2^1 + \tilde{Y}_2^1 + \dots + Y_R^1 + \tilde{Y}_R^1 \middle| \mathbf{E}_{011} \right\} \quad (27)$$

$$\geq \mathbb{E} \left\{ Y_1^1 + Y_2^1 + \dots + Y_R^1 \middle| \mathbf{E}_{011} \right\} \quad (28)$$

$$= \mathbb{E} \left\{ Y_1^1 + \sum_{i=1}^{Z_1^2} Y_2^1(i) + \dots + \sum_{i=1}^{Z_{R-1}^2} Y_R^1(i) \middle| \mathbf{E}_{011} \right\}. \quad (29)$$

We need to note that for each r , conditioning on $\mathbf{E}_{011}$, $\vec{Z}_{r-1}, \vec{Y}_{r-1}$, and $Z_{r-1}^2 > 0$, the random variables $Y_r^1(i)$, $i = 1, \dots, Z_{r-1}^2$, are i.i.d. and have the same distribution as $Y_1^1(1)$, and are independent from Z_{r-1}^2 .

Let $\vec{Y}_r = (Y_1^1, Y_2^1, \dots, Y_r^1, Y_1^2, Y_2^2, \dots, Y_r^2)$, $r = 1, \dots, R$, and let $\vec{Z}_r = (Z_1^1, Z_2^1, \dots, Z_r^1, Z_1^2, Z_2^2, \dots, Z_r^2)$, $r = 1, \dots, R$.

$$\begin{aligned} & \mathbb{E} \left\{ Y_1^1 + \sum_{i=1}^{Z_1^2} Y_2^1(i) + \dots + \sum_{i=1}^{Z_{R-1}^2} Y_R^1(i) \middle| \mathbf{E}_{011} \right\} \\ &= \mathbb{E} \left\{ \mathbb{E} \left\{ Y_1^1 + \sum_{i=1}^{Z_1^2} Y_2^1(i) + \dots + \sum_{i=1}^{Z_{R-1}^2} Y_R^1(i) \middle| \vec{Y}_{R-1}, \vec{Z}_{R-1} \right\} \middle| \mathbf{E}_{011} \right\} \\ &= \mathbb{E} \left\{ Y_1^1 + \sum_{i=1}^{Z_1^2} Y_2^1(i) + \dots + \sum_{i=1}^{Z_{R-2}^2} Y_{R-1}^1(i) \middle| \mathbf{E}_{011} \right\} \\ &\quad + \mathbb{E} \left\{ \mathbb{E} \left\{ \sum_{i=1}^{Z_{R-1}^2} Y_R^1(i) \middle| \vec{Y}_{R-1}, \vec{Z}_{R-1} \right\} \middle| \mathbf{E}_{011} \right\} \\ &= \mathbb{E} \left\{ Y_1^1 + \sum_{i=1}^{Z_1^2} Y_2^1(i) + \dots + \sum_{i=1}^{Z_{R-2}^2} Y_{R-1}^1(i) \middle| \mathbf{E}_{011} \right\} \\ &\quad + \mathbb{E} \{ Z_{R-1}^2 | \mathbf{E}_{011} \} \mathbb{E} \{ Y_1^1(1) | \mathbf{E}_{011} \}, \end{aligned} \quad (30)$$

where the last equality follows from the conditional independence between Z_{R-1}^2 and $Y_1^1(k)$, $k = 1$ to Z_{R-1}^2 in the event

$Z_{R-1}^2 > 0$. By iteratively applying the procedures as in (30) to (31), we have

$$\begin{aligned} & \mathbb{E} \left\{ Y_1^1 + \sum_{i=1}^{Z_1^2} Y_2^1(i) + \dots + \sum_{i=1}^{Z_{R-1}^2} Y_R^1(i) \middle| \mathbf{E}_{011} \right\} \\ &= \mathbb{E}\{Y_1^1(1)|\mathbf{E}_{011}\} + \mathbb{E}\{Z_1^2|\mathbf{E}_{011}\}\mathbb{E}\{Y_1^1(1)|\mathbf{E}_{011}\} + \dots \\ & \quad + \mathbb{E}\{Z_{R-1}^2|\mathbf{E}_{011}\}\mathbb{E}\{Y_1^1(1)|\mathbf{E}_{011}\} \\ &= \mathbb{E}\{Y_1^1|\mathbf{E}_{011}\}\mathbb{E}\{1 + Z_1^2 + \dots + Z_{R-1}^2|\mathbf{E}_{011}\}. \end{aligned} \quad (32)$$

To quantify the term $\mathbb{E}\{Y_1^1|\mathbf{E}_{011}\}$, we notice that for the 1st round transmission with L_{011} , in every slot of coded transmission one packet leaves L_{100} with probability $1 - (1-p)^2$ since any one of users 2 and 3 receives the coded transmission will lead to one packet leaving L_{100} . In addition, the number of average transmissions before the packet in L_{011} leaves L_{011} is $\frac{1}{p}$. Again by similar derivation steps as in (30) to (31), we have $\mathbb{E}\{Y_1^1(1)|\mathbf{E}_{011}\} = \frac{1}{p}(2p - p^2)$.

We now quantify the expectations $\mathbb{E}\{Z_r^1|\mathbf{E}_{011}\}$ and $\mathbb{E}\{Z_r^2|\mathbf{E}_{011}\}$ for $r = 1, 2, \dots$. Consider Z_1^1 first. For the 1st round transmission with L_{011} , in every slot of coded transmission one packet leaves L_{100} and enters L_{110} with probability $p(1-p)$ since such transition happens when user 2 receives the packet but not user 3. By similar reasons as in the previous steps, $\mathbb{E}\{Z_1^1(1)|\mathbf{E}_{011}\} = \frac{1}{p}p(1-p) = 1-p$. We now consider Z_1^2 and we have

$$\begin{aligned} \mathbb{E}\{Z_1^2|\mathbf{E}_{011}\} &= \mathbb{E} \left\{ \sum_{i=1}^{Z_1^1} Z_1^2(i) \middle| \mathbf{E}_{011} \right\} \\ &= \mathbb{E}\{Z_1^1|\mathbf{E}_{011}\}\mathbb{E}\{Z_1^2(1)|\mathbf{E}_{011}, Z_1^1 > 0\} \end{aligned} \quad (33)$$

$$= \left(\frac{1}{p}p(1-p)\right) \left(\frac{1}{p}p(1-p)\right) = (1-p)^2, \quad (34)$$

where (33) follows from that given Z_1^1 , the random variables $Z_1^2(1)$ to $Z_1^2(Z_1^1)$ are i.i.d., and (34) follows from that conditioning on $Z_1^1 > 0$, $Z_1^2(1)$ has the same distribution as Z_1^1 . The expression of $\mathbb{E}\{Z_r^1|\mathbf{E}_{011}\}$ and $\mathbb{E}\{Z_r^2|\mathbf{E}_{011}\}$ for general r values can be obtained iteratively by

$$\begin{aligned} \mathbb{E}\{Z_r^1|\mathbf{E}_{011}\} &= \mathbb{E} \left\{ \sum_{i=1}^{Z_{r-1}^2} Z_r^1(i) \middle| \mathbf{E}_{011} \right\} \\ &= \mathbb{E}\{Z_{r-1}^2|\mathbf{E}_{011}\}\mathbb{E}\{Z_r^1(1)|\mathbf{E}_{011}, Z_{r-1}^2 > 0\} \\ &= (1-p)^{2(r-1)} \left(\frac{1}{p}p(1-p)\right) = (1-p)^{2r-1}, \end{aligned}$$

and

$$\begin{aligned} \mathbb{E}\{Z_r^2|\mathbf{E}_{011}\} &= \mathbb{E} \left\{ \sum_{i=1}^{Z_r^1} Z_r^2(i) \middle| \mathbf{E}_{011} \right\} \\ &= \mathbb{E}\{Z_r^1|\mathbf{E}_{011}\}\mathbb{E}\{Z_r^2(1)|\mathbf{E}_{011}, Z_r^1 > 0\} \\ &= (1-p)^{2r-1} \left(\frac{1}{p}p(1-p)\right) = (1-p)^{2r}. \end{aligned}$$

By (32), we then have

$$\begin{aligned} & \mathbb{E} \left\{ Y_1^1 + \sum_{i=1}^{Z_1^2} Y_2^1(i) + \dots + \sum_{i=1}^{Z_{R-1}^2} Y_R^1(i) \middle| \mathbf{E}_{011} \right\} \\ &= (2-p) \left(1 + (1-p)^2 + (1-p)^4 + \dots + (1-p)^{2R-2}\right). \end{aligned} \quad (35)$$

The results in (35) can help to show the negative drift for $L_{100}(t_1, n-1)$. However, recall that the above analysis assume that there is no packet expiration and the lengths of $L_{001}(t_1, n-1)$ and $L_{100}(t_1, n-1)$ are infinite. For the following, we remove the latter assumption by assuming $|L_{100}(t_1, n-1)| = \overline{B}^1$, $|L_{001}(t_1, n-1)| = \overline{B}^2$ for some finitely large $\overline{B}^1$ and $\overline{B}^2$. Since now we only have finite $|L_{100}(t_1, n-1)|$ and $|L_{001}(t_1, n-1)|$, with some strictly positive probability that one of these two lists may be depleted before finishing R rounds of coded transmission, as was discussed previously.

Let $A_{\tilde{k}}$ denote the event that the $\tilde{k}$ -th packet that leaves L_{001} actually enters L_{011} , and let $1_{A_{\tilde{k}}}$ denote the indicator function

for the event $A_{\bar{k}}$. We then have

$$\sum_{i=1}^{R-1} Z_i^2 + \tilde{Z}_i^2 = \min\left(\sum_{i=1}^{R-1} Y_i^2 + \tilde{Y}_i^2, \overline{B^2}\right) \sum_{\bar{k}=1} 1_{A_{\bar{k}}}, \quad (36)$$

where the left-hand side corresponds to counting the packets leaving L_{001} and entering L_{011} by “rounds” and the right-hand side corresponds to counting in a per-packet basis. The minimum operation $\min(\cdot, \overline{B^2})$ follows from the new assumption that we only have a finite length of $L_{100}(t_1, n-1)$. Then when both $|L_{100}(t_1, n-1)| = \overline{B^1}$, $|L_{001}(t_1, n-1)| = \overline{B^2}$ are finite, the total number of packets leaving L_{100} (conditioning on that the uncoded packet n enters L_{011}) can be expressed by

$$S_{100|011} = \min \left(\sum_{l=1}^{1 + \sum_{\bar{k}=1}^{\min(\sum_{i=1}^{R-1} Y_i^2 + \tilde{Y}_i^2, \overline{B^2})} 1_{A_{\bar{k}}} + Z^3} Y^1(l), \overline{B^1} \right), \quad (37)$$

where the operation $\min(\cdot, \overline{B^1})$ ensures that the largest number of packets (in the the current time scope) that can leave list L_{100} is at most $\overline{B^1}$. As discussed previously, the packets leaving L_{100} are denoted by a finite sequence of random variables $Y_1^1, \tilde{Y}_1^1, Y_2^1(1), \dots, Y_2^1(Z_1^2), \tilde{Y}_2^1(1), \dots, Y_R^1(1), \dots, Y_R^1(Z_{R-1}^2)$. Similar to (36), we can relabel the random variables by their order in the overall sequence as $Y^1(l)$, $l = 1, \dots, 1 + \sum_{i=1}^{R-1} Z_i^2 + \sum_{i=1}^{R-1} \tilde{Z}_i^2 + Z^3$, where $1 + \sum_{i=1}^{R-1} Z_i^2$ of the random variables are previously denoted by $Y_r^1(k)$ for some $r = 1, \dots, R$ and some $k = 1, \dots, Z_{r-1}^2$; $\sum_{i=1}^{R-1} \tilde{Z}_i^2$ of the random variables are previously denoted by $\tilde{Y}_r^1(k)$ for some $r = 2, \dots, R$ and some $k = 1, \dots, \tilde{Z}_{r-1}^2$; and the remaining Z^3 random variable corresponds to mixing a L_{100} packet with a L_{011} packet that previously left L_{010} and entered L_{011} . By taking the minimum of $\overline{B^1}$ and the sum of all such $Y^1(l)$ random variables, we thus obtain the number of packets leaving L_{100} . The superscript of the summation of $Y^1(l)$ is then rewritten by (36).

Based on (37), we can further lower bound $S_{100|011}$

$$S_{100|011} \geq \min \left(\sum_{l=1}^{1 + \sum_{\bar{k}=1}^{\min(\sum_{i=1}^{R-1} Y_i^2, \overline{B^2})} 1_{A_{\bar{k}}}} Y^1(l), \overline{B^1} \right). \quad (38)$$

Next we are going to show that when $\overline{B^1}, \overline{B^2} \rightarrow \infty$, we have

$$\begin{aligned} & \mathbb{E} \left\{ \min \left(\sum_{l=1}^{1 + \sum_{\bar{k}=1}^{\min(\sum_{i=1}^{R-1} Y_i^2, \overline{B^2})} 1_{A_{\bar{k}}}} Y^1(l), \overline{B^1} \right) \middle| \mathbb{E}_{011} \right\} \\ & \rightarrow \mathbb{E} \left\{ \sum_{l=1}^{1 + \sum_{i=1}^{R-1} Z_i^2} Y^1(l) \middle| \mathbb{E}_{011} \right\} \end{aligned}$$

For any integer $b > 0$, let

$$G_b = \min \left(\sum_{l=1}^{1 + \sum_{\bar{k}=1}^{\min(\sum_{i=1}^{R-1} Y_i^2, b)} 1_{A_{\bar{k}}}} Y^1(l), b \right).$$

One can quickly verify that for any realization in the sample space,

$$G_{b+1} = \min \left(\sum_{l=1}^{1 + \sum_{\bar{k}=1}^{\min(\sum_{i=1}^{R-1} Y_i^2, b+1)} 1_{A_{\bar{k}}}} Y^1(l), b+1 \right) \geq G_b.$$

Since the sequence $\{G_b\}$ is monotonically increasing, and converges to

$$G \triangleq \sum_{l=1}^{1+\sum_{k=1}^{R-1} Y_k^2} 1_{A_k} Y^1(l) = \sum_{l=1}^{1+\sum_{i=1}^{R-1} Z_i^2} Y^1(l), \quad (39)$$

by the monotone convergence theorem, we have

$$\lim_{b \rightarrow \infty} \mathbb{E}\{G_b | \mathbb{E}_{011}\} \rightarrow \mathbb{E}\left\{ \sum_{l=1}^{1+\sum_{i=1}^{R-1} Z_i^2} Y^1(l) \middle| \mathbb{E}_{011} \right\}. \quad (40)$$

By similar arguments as in the previous derivation, we also have

$$\begin{aligned} & \mathbb{E}\left\{ \sum_{l=1}^{1+\sum_{i=1}^{R-1} Z_i^2} Y^1(l) \middle| \mathbb{E}_{011} \right\} \\ &= \mathbb{E}\left\{ 1 + \sum_{i=1}^{R-1} Z_i^2 \middle| \mathbb{E}_{011} \right\} \mathbb{E}\{Y_1^1 | \mathbb{E}_{011}\} \\ &= \mathbb{E}\{Y_1^1 | \mathbb{E}_{011}\} \mathbb{E}\{1 + Z_1^2 + \dots + Z_{R-1}^2 | \mathbb{E}_{011}\} \\ &= \text{Equation (35)}. \end{aligned}$$

As a result, for any $\epsilon^* > 0$, $\exists B_3 > 0$, such that

$$\begin{aligned} & \mathbb{E}\{G_{B_3} | \mathbb{E}_{011}\} \\ & > (1 - \epsilon^*)(1 + (1 - p)^2 + (1 - p)^4 + \dots + (1 - p)^{2R-2})(2 - p). \end{aligned}$$

Note that the monotonicity argument can also be applied to individual $\overline{B_1}$ and $\overline{B_2}$, respectively. As a result, for any $\overline{B^1}, \overline{B^2} > B_3$,

$$\begin{aligned} & \mathbb{E}\left\{ \min \left(\sum_{l=1}^{1+\sum_{k=1}^{\min(\sum_{i=1}^{R-1} Y_i^2, \overline{B^2})} 1_{A_k}} y^1(l, \overline{B^1}) \right) \middle| \mathbb{E}_{011} \right\} \\ & > (1 - \epsilon^*)(1 + (1 - p)^2 + (1 - p)^4 + \dots + (1 - p)^{2R-2})(2 - p). \end{aligned} \quad (41)$$

The above discussion focuses on the expected changes of $L_{100}(t_1, n)$ when the uncoded packet n enters L_{011} . Similar arguments can be made to discuss the case when the first the uncoded packet enters L_{110} . More explicitly, given that the uncoded packet enters list L_{110} , let $S_{100|110}$ denote the number of packets leaving list L_{100} within R rounds of transmission that alternates between coding groups (L_{001}, L_{110}) (since the first opportunity to form IDNC packet will be (L_{001}, L_{110})) and (L_{100}, L_{011}) . We then have for any $\epsilon^* > 0$, $\exists B_4 > 0$, such that when $\overline{B^2}, \overline{B^1} > B_4$, we have

$$\begin{aligned} & \mathbb{E}\{S_{100|110} | \mathbb{E}_{110}\} \\ & > (1 - \epsilon^*)(2 - p) \left((1 - p) + (1 - p)^3 + \dots + (1 - p)^{2R-1} \right). \end{aligned} \quad (42)$$

Note that the values of B_3 and B_4 depends on R , the number of rounds we are considering. For any ϵ^* in the interval $(0, \frac{1}{2-p})$, we choose a sufficiently large R such that

$$R > \frac{1}{2} \left(1 + \log_{1-p} \left(1 - \frac{1-p}{(2-p)(1-\epsilon^*)} \right) \right). \quad (43)$$

Define $B_0 = \max(B_3, B_4)$ and define $\Lambda = \{(\alpha, \beta) | \alpha < B_0 \text{ or } \beta < B_0\}$ and its complement $\Lambda^c = (R^+)^2 \setminus \Lambda$. By (41) and (42), we then have that

$$\begin{aligned} & \mathbb{E}\{|L_{100}(t_1, n)| - |L_{100}(t_1, n-1)| \mid (|L_{001}(t_1, n-1)|, |L_{100}(t_1, n-1)|) \in \Lambda^c, q(t_1) > B\} \\ & < -\frac{p_{011}}{1-p_{000}} \mathbb{E}\{S_{100|011} | \mathbb{E}_{011}\} - \frac{p_{110}}{1-p_{000}} \mathbb{E}\{S_{100|110} | \mathbb{E}_{110}\} + \frac{p_{100}}{1-p_{000}} \end{aligned} \quad (44)$$

$$\leq -p^2(1-p)(1-\epsilon^*)(2-p) \frac{1-(1-p)^{2R-1}}{1-(1-p)} + p(1-p)^2 \quad (45)$$

$$< 0, \quad (46)$$

where $\frac{p_{011}}{1-p_{000}}$ is the probability that when the uncoded packet n is received by at least one user, it actually enters list L_{011} ,

$\frac{p_{110}}{1-p_{000}}$ is the probability that it actually enters list L_{110} , and $\frac{p_{100}}{1-p_{000}}$ is the probability that it actually enters list L_{100} . Since our derivation does not depend on the order of the two terms $|L_{001}(t_1, n-1)|$ and $|L_{100}(t_1, n-1)|$, we also have

$$\mathbb{E}\{|L_{100}(t_1, n)| - |L_{100}(t_1, n-1)| \mid (|L_{001}(t_1, n-1)|, |L_{100}(t_1, n-1)|) \in \Lambda^c, |L_{001}(t_1, n-1)| \leq |L_{100}(t_1, n-1)|, q(t_1) > B\} < 0, \quad (47)$$

$$\mathbb{E}\{|L_{100}(t_1, n)| - |L_{100}(t_1, n-1)| \mid (|L_{001}(t_1, n-1)|, |L_{100}(t_1, n-1)|) \in \Lambda^c, |L_{001}(t_1, n-1)| > |L_{100}(t_1, n-1)|, q(t_1) > B\} < 0. \quad (48)$$

The above shows the negative drift from $L_{100}(t_1, n-1)$ to $L_{100}(t_1, n)$. By symmetry, the size of the list $L_{001}(t_1, n-1)$ to $L_{001}(t_1, n)$ also has a negative drift. For the following, we will show that the minimum of the two lists $V(t_1, n) = \min(|L_{001}(t_1, n)|, |L_{100}(t_1, n)|)$ also has a negative drift when both $|L_{001}(t_1, n-1)|$ and $|L_{100}(t_1, n-1)|$ are large. Continue from the previous discussion, we still assume that there is no packet expiration and will consider the situation with packet expiration in the end of this section. To prove that $V(t_1, n-1)$ has a negative drift when both $|L_{001}(t_1, n-1)|$ and $|L_{100}(t_1, n-1)|$ are large, we need the following lemma.

Lemma 12. Recall that $\mathcal{N}_1 \triangleq n(t_1)$ is the packet index at time t_1 . Define a sequence of 2-dimensional vectors $\mathbf{X} = \{\mathbf{X}_n = (|L_{001}(t_1, n)|, |L_{100}(t_1, n)|) : \forall n \geq \mathcal{N}_1\}$. Let $T_0 = \mathcal{N}_1$. Define $T_1 = \inf\{n : \mathbf{X}_n \in \Lambda^c, n \geq \mathcal{N}_1\}$, $T_2 = \inf\{n : n \geq T_1, \mathbf{X}_n \in \Lambda\}$, ..., $T_{2i-1} = \inf\{n : n \geq T_{2i-2}, \mathbf{X}_n \in \Lambda^c\}$, $T_{2i} = \inf\{n : n \geq T_{2i-1}, \mathbf{X}_n \in \Lambda\}$. Namely, T_{2i} and T_{2i+1} are the stopping times that the random vector $\mathbf{X}_n$ moves in and out of Λ , respectively. Then $\{V(t_1, (l + T_{2i-1}) \wedge T_{2i}) : \forall l = 1, 2, \dots\}$ is a super martingale with respect to the index l .

Proof: The proof for Lemma 12 is as follows: For a fixed $i \geq 1$, T_{2i} is a stopping time, so by (48) we can show for $n-1 \geq T_{2i-1}$,

$$\begin{aligned} & \mathbb{E}\{|L_{100}(t_1, n \wedge T_{2i})| \mid |L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})|, |L_{001}(t_1, (n-1) \wedge T_{2i})|, \\ & \quad |L_{100}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}\} \\ &= \mathbb{E}\{|L_{100}(t_1, n)| \mid |L_{001}(t_1, n-1)|, |L_{100}(t_1, n-1)|, |L_{100}(t_1, n-1)| < |L_{001}(t_1, n-1)|, q(t_1) > B, T_{2i} > n-1 \geq T_{2i-1}\} \\ & \quad \times \mathbb{P}(T_{2i} > n-1 \mid |L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})|, \\ & \quad |L_{001}(t_1, (n-1) \wedge T_{2i})|, |L_{100}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}) \\ & \quad + \mathbb{E}\{|L_{100}(t_1, T_{2i})| \mid |L_{001}(t_1, T_{2i})|, |L_{100}(t_1, T_{2i})|, |L_{100}(t_1, T_{2i})| < |L_{001}(t_1, T_{2i})|, q(t_1) > B, T_{2i} \leq n-1, n-1 \geq T_{2i-1}\} \\ & \quad \times \mathbb{P}(T_{2i} \leq n-1 \mid |L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})|, \\ & \quad |L_{001}(t_1, (n-1) \wedge T_{2i})|, |L_{100}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}) \\ &< \mathbb{E}\{|L_{100}(t_1, n-1)| \mid |L_{001}(t_1, n-1)|, |L_{100}(t_1, n-1)|, |L_{100}(t_1, n-1)| < |L_{001}(t_1, n-1)|, \\ & \quad q(t_1) > B, T_{2i} > n-1, n-1 \geq T_{2i-1}\} \\ & \quad \times \mathbb{P}(T_{2i} > n-1 \mid |L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})|, \\ & \quad |L_{001}(t_1, (n-1) \wedge T_{2i})|, |L_{100}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}) \\ & \quad + \mathbb{E}\{|L_{100}(t_1, T_{2i})| \mid |L_{001}(t_1, T_{2i})|, |L_{100}(t_1, T_{2i})|, |L_{100}(t_1, T_{2i})| < |L_{001}(t_1, T_{2i})|, q(t_1) > B, T_{2i} \leq n-1, n-1 \geq T_{2i-1}\} \\ & \quad \times \mathbb{P}(T_{2i} \leq n-1 \mid |L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})|, \\ & \quad |L_{001}(t_1, (n-1) \wedge T_{2i})|, |L_{100}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}) \\ &= \mathbb{E}\{|L_{100}(t_1, (n-1) \wedge T_{2i})| \mid |L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})|, |L_{001}(t_1, (n-1) \wedge T_{2i})|, \\ & \quad |L_{100}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, T_{2i} > n-1, n-1 \geq T_{2i-1}\} \\ & \quad \times \mathbb{P}(T_{2i} > n-1 \mid |L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})|, \\ & \quad |L_{001}(t_1, (n-1) \wedge T_{2i})|, |L_{100}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}) \\ & \quad + \mathbb{E}\{|L_{100}(t_1, (n-1) \wedge T_{2i})| \mid |L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})|, |L_{001}(t_1, (n-1) \wedge T_{2i})|, \\ & \quad |L_{100}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, T_{2i} \leq n-1, n-1 \geq T_{2i-1}\} \\ & \quad \times \mathbb{P}(T_{2i} \leq n-1 \mid |L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})|, \\ & \quad |L_{001}(t_1, (n-1) \wedge T_{2i})|, |L_{100}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}) \\ &= \mathbb{E}\{|L_{100}(t_1, (n-1) \wedge T_{2i})| \mid |L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})|, |L_{001}(t_1, (n-1) \wedge T_{2i})|, \\ & \quad |L_{100}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}\} \\ &= |L_{100}(t_1, (n-1) \wedge T_{2i})|, \end{aligned} \quad (49)$$

where (49) is obtained from the inequality in (48). So we have proven that $|L_{100}(t_1, (n-1) \wedge T_{2i})|$ has a negative drift conditioning on $n-1 \geq T_{2i-1}$, $|L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})|$, $|L_{001}(t_1, (n-1) \wedge T_{2i})|$, and $|L_{100}(t_1, (n-1) \wedge T_{2i})|$.

$1) \wedge T_{2i})|$.

Similarly by (47) we have

$$\begin{aligned} & \mathbb{E}\{|L_{100}(t_1, n \wedge T_{2i})| \mid |L_{100}(t_1, (n-1) \wedge T_{2i})| \geq |L_{001}(t_1, (n-1) \wedge T_{2i})|, |L_{001}(t_1, (n-1) \wedge T_{2i})|, \\ & \quad |L_{100}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}\} \\ & < |L_{100}(t_1, (n-1) \wedge T_{2i})|. \end{aligned}$$

By symmetry, we can also obtain the similar results for $L_{001}(t_1, n \wedge T_{2i})$. Jointly, we thus have

$$\begin{aligned} & \mathbb{E}\{V(t_1, n \wedge T_{2i}) \mid |L_{100}(t_1, (n-1) \wedge T_{2i})|, |L_{001}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}\} \\ & = \mathbb{E}\{V(t_1, n \wedge T_{2i}) \mid |L_{100}(t_1, (n-1) \wedge T_{2i})|, |L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}\} \\ & \quad \times \mathbb{P}(|L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})| \mid q(t_1) > B, n-1 \geq T_{2i-1}) \\ & + \mathbb{E}\{V(t_1, n \wedge T_{2i}) \mid |L_{001}(t_1, (n-1) \wedge T_{2i})|, |L_{100}(t_1, (n-1) \wedge T_{2i})| \geq |L_{001}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}\} \\ & \quad \times \mathbb{P}(|L_{100}(t_1, (n-1) \wedge T_{2i})| \geq |L_{001}(t_1, (n-1) \wedge T_{2i})| \mid q(t_1) > B, n-1 \geq T_{2i-1}) \\ & < \mathbb{E}\{|L_{100}(t_1, n \wedge T_{2i})| \mid |L_{100}(t_1, (n-1) \wedge T_{2i})|, |L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}\} \\ & \quad \times \mathbb{P}(|L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})| \mid q(t_1) > B, n-1 \geq T_{2i-1}) \\ & + \mathbb{E}\{|L_{100}(t_1, n \wedge T_{2i})| \mid |L_{001}(t_1, (n-1) \wedge T_{2i})|, |L_{100}(t_1, (n-1) \wedge T_{2i})| \geq |L_{001}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}\} \\ & \quad \times \mathbb{P}(|L_{100}(t_1, (n-1) \wedge T_{2i})| \geq |L_{001}(t_1, (n-1) \wedge T_{2i})| \mid q(t_1) > B, n-1 \geq T_{2i-1}) \\ & < |L_{100}(t_1, (n-1) \wedge T_{2i})| (\mathbb{P}(|L_{100}(t_1, (n-1) \wedge T_{2i})| < |L_{001}(t_1, (n-1) \wedge T_{2i})| \mid q(t_1) > B, n-1 \geq T_{2i-1}) \\ & \quad + \mathbb{P}(|L_{100}(t_1, (n-1) \wedge T_{2i})| \geq |L_{001}(t_1, (n-1) \wedge T_{2i})| \mid q(t_1) > B, n-1 \geq T_{2i-1})) \\ & = |L_{100}(t_1, (n-1) \wedge T_{2i})|. \end{aligned}$$

By symmetry, we have

$$\mathbb{E}\{V(t_1, n \wedge T_{2i}) \mid |L_{100}(t_1, (n-1) \wedge T_{2i})|, |L_{001}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}\} < |L_{001}(t_1, (n-1) \wedge T_{2i})|.$$

Combining the above two inequalities, we thus have

$$\mathbb{E}\{V(t_1, n \wedge T_{2i}) \mid |L_{100}(t_1, (n-1) \wedge T_{2i})|, |L_{001}(t_1, (n-1) \wedge T_{2i})|, q(t_1) > B, n-1 \geq T_{2i-1}\} < V(t_1, (n-1) \wedge T_{2i}), \quad (50)$$

which means $\{V(t_1, (l + T_{2i-1}) \wedge T_{2i})\}$ is a super martingale for $l \in \mathcal{N}$ ($\mathcal{N}$ is the set of all positive integers). The proof for Lemma 12 is complete. $\blacksquare$

Lemma 12 implies that

$$\begin{aligned} & \mathbb{E}\{V(t_1, n \wedge T_{2i}) \mid q(t_1) > B, n-1 \geq T_{2i-1}\} < \mathbb{E}\{V(t_1, (n-1) \wedge T_{2i}) \mid q(t_1) > B, n-1 \geq T_{2i-1}\} \\ & \leq \mathbb{E}\{V(t_1, T_{2i-1}) \mid q(t_1) > B, n-1 \geq T_{2i-1}\} = B_0. \end{aligned} \quad (51)$$

The inequality in (51) is due to the negative drift shown in (50). For any $\epsilon'_1 > 0$, let $\bar{B}_1 = \frac{6B_0}{\epsilon'_1}$. Then, by the Markov inequality, for $\forall B_6 > \bar{B}_1$, we have

$$\mathbb{P}(V(t_1, n) > \epsilon'_1 B_6/2 \mid n \in \cup_{i \in \mathcal{N}} [T_{2i-1}, T_{2i}), q(t_1) > B) \leq \frac{2B_0}{\epsilon'_1 \bar{B}_1} = \epsilon'_1/3.$$

Also, when $\mathbf{X}_n \in \Lambda$, that is when $n \in \cup_{i \in \mathcal{N}} [T_{2i}, T_{2i+1})$, we always have

$$\mathbb{P}(V(t_1, n) > \epsilon'_1 B_6/2 \mid n \in \cup_{i \in \mathcal{N}} [T_{2i}, T_{2i+1}), q(t_1) > B) = 0.$$

Then by the total probability theorem, we have

$$\begin{aligned} & \mathbb{P}(V(t_1, n) > \epsilon'_1 B_6/2 \mid q(t_1) > B) \\ & = \mathbb{P}(V(t_1, n) > \epsilon'_1 B_6/2 \mid \exists i, \text{ s.t. } n \in [T_{2i-1}, T_{2i}), q(t_1) > B) \mathbb{P}(\exists i, \text{ s.t. } n \in [T_{2i-1}, T_{2i}) \mid q(t_1) > B) \\ & \quad + \mathbb{P}(V(t_1, n) > \epsilon'_1 B_6/2 \mid \exists i, \text{ s.t. } n \in [T_{2i}, T_{2i+1}), q(t_1) > B) \mathbb{P}(\exists i, \text{ s.t. } n \in [T_{2i}, T_{2i+1}) \mid q(t_1) > B) \\ & \leq \epsilon'_1/3, \quad \forall n \geq \mathcal{N}_1, \end{aligned}$$

which implies that

$$\mathbb{P}(|L_{001}(t_1, n)| \geq \epsilon'_1 B_6/2, |L_{010}(t_1, n)| \geq \epsilon'_1 B_6/2 \mid q(t_1) > B) < \epsilon'_1/3, \text{ for all } n \geq \mathcal{N}_1.$$

Here n denotes the index of super slots. We then note that during the entire super slot, the length of each list on the outer circle can increase at most by 1. Suppose the very last time slot of super slot $\mathcal{N}_1$ is time slot $t_{\text{end}}(\mathcal{N}_1)$. Then we choose

$B_5 = t_{\text{end}}(\mathcal{N}_1) - t_1$. For all $t > t_1 + B_5$, $B_6 > \bar{B}_1 > 0$, by choosing $B_7 = \frac{\epsilon'_1 B_6 + 2}{\epsilon'_1 \lambda}$ we then have

$$P(|L_{001}(t_1, t)| \geq \epsilon'_1 \lambda B_7 / 2, |L_{010}(t_1, t)| \geq \epsilon'_1 \lambda B_7 / 2 | q(t_1) > B) < \epsilon'_1 / 3 \quad (52)$$

Recall that we use “the outer circle” to denote the lists L_{100} , L_{001} , and L_{010} . Since (52) is true for any two lists on the outer circle, by the union bound we have for all $B > B_1 = \max(B_5, B_7)$,

$$P\left(\text{at least two } \mathbf{v}\text{'s in } \{100, 010, 001\} \text{ have } |L_{\mathbf{v}}(t_1, t_1 + B)| \geq \frac{\epsilon'_1 \lambda B}{2} | q(t_1) > B\right) \leq \epsilon'_1. \quad (53)$$

In the previous discussions, we only talk about the case without considering packet expirations. For the following, we will prove why $V(t_1, (n-1) \wedge T_{2i})$ (conditioning on $n-1 \geq T_{2i-1}$) still has a negative drift even when considering packet expiration. Since our previous analysis only depends on $V(t_1, (n-1) \wedge T_{2i})$ having a negative drift, the proof of Proposition 5 thus follows verbatim.

Define $\tilde{V}(t_1, n-1)$ the same way as $V(t_1, n-1)$ except that now we consider packet expiration. First, we will show $\tilde{V}(t_1, n-1)$ has a stronger negative drift than that of $V(t_1, n-1)$ when $(L_{100}(t_1, n-1), L_{001}(t_1, n-1))$ are in Λ^c . The intuition behind is that expiration will help draining more packets from $L_{100}(t_1, n-1)$ and $L_{001}(t_1, n-1)$ when compared to a non-expiration scenario.

Recall that conditioning on $(L_{100}(t_1, n-1), L_{001}(t_1, n-1))$ being in Λ^c , both L_{110} and L_{011} are completely empty at the beginning of super slot n . Otherwise there will be opportunity to form IDNC packet either (L_{100}, L_{011}) or (L_{001}, L_{110}) and the super time slot of the previous packet $(n-1)$ will not end. Suppose when the BS sends the uncoded packet n in the beginning of super slot n , this uncoded packet enters list L_{001} (resp. L_{100}). The length of $L_{100}(t_1, n)$ (resp. $L_{001}(t_1, n)$) is increased by 1 since the packet n (recall that $n \geq \mathcal{N}_1$) is within the current time scope and is counted toward $L_{100}(t_1, n)$ (resp. $L_{001}(t_1, n)$). Therefore, considering expiration will not affect the increment from $L_{100}(t_1, n-1)$ to $L_{100}(t_1, n)$ in the scenario when the uncoded packet n enters L_{100} .

Next we will show that with expiration, the number of packets leaving $L_{100}(t_1, n)$ (resp. $L_{001}(t_1, n)$) due to packet n entering L_{011} or L_{110} will not decrease.

Suppose the transmitted uncoded packet enters L_{011} or L_{110} , and triggers coded transmission. Next we consider the effect of $\tilde{V}(t_1, n)$ with packet expiration.

Case 1: In the end of super time slot n , neither $L_{100}(t_1, n)$ nor $L_{001}(t_1, n)$ is empty. Since we always choose the youngest packet from the outer circle for coded transmission, it means that all the packets leaving out of L_{100} and L_{001} during the current super time slot are “of the current time-scope” (i.e., those packets will not expire between time slot t_1 and λn). Therefore, the “seeds” of the chain effects (those packets leaving from L_{100} , L_{001} and entering L_{011} , L_{110}) will not expire during the current super time slot. So the packet departure caused by the coded transmission is not affected by the expiration. Since during our previous analysis, we only quantify the amount of departure caused by those seed packets, therefore, the absolute value of the decrement can be lower bounded in the same way as our previous analysis.

Case 2: In the end of super time slot n , at least one of $L_{100}(t_1, n)$ and $L_{001}(t_1, n)$ becomes empty. In this case $\tilde{V}(t_1, n)$, the minimum of $|L_{100}(t_1, n)|$ and $|L_{001}(t_1, n)|$, is simply zero. As a result, we have $\tilde{V}(t_1, n) \leq V(t_1, n)$. From the above analysis, the negative drift of $\tilde{V}(t_1, n-1)$ is no smaller than that of $V(t_1, n-1)$ in our previous analysis. The proof is complete. $\blacksquare$

APPENDIX B PROOF FOR PROPOSITION 6

Proof: In this proof, we use the notation of super time slot as defined in the proof of Proposition 5. Note that at the beginning of one super slot, no coded packet can be mixed together and an uncoded packet is sent. Therefore, in each coding group, at least one of its constituent lists must be empty. Since the coding groups (L_{001}, L_{110}) , (L_{010}, L_{101}) , and (L_{100}, L_{011}) do not share any common list, there must be at least three lists that are empty in the beginning of the super time slot. As a result, there are at most three lists that are non-empty at the beginning of the super time slot. Moreover, the set of non-empty lists must not be a superset of any of the coding group. We use $\text{RC}_j(t_1, n)$ to denote the total number of coding opportunities in the current time scope involving user j at the end of super slot n . For example, $\text{RC}_3(t_1, n) = |L_{100}(t_1, n)| + |L_{110}(t_1, n)| + |L_{010}(t_1, n)|$. Note that in one super slot, some packet may leave one list and join another one, for example, a packet may leave L_{100} and join L_{110} . We also observe that such packet relocation can only happen among those lists whose packets are the same user's coding opportunities. For example, only packets from L_{100} and L_{010} can join L_{110} , and packets from these three lists are all coding opportunities involving user 3.

Since there is only one uncoded packet transmitted by the BS in one super slot, which potentially may increase the number of coding opportunities, and since all other coded packet transmission will either destroy coding opportunities or cause packet relocation within the lists that are considered as the coding opportunities of the same user, the total number of coding opportunities in the current time scope involving user j during the super slot n , is upper bounded by $\text{RC}_j(t_1, t) \leq \text{RC}_j(t_1, n-1) + 1$ for all time slot t in the super time slot n , for $j = 1, 2, 3$, respectively. Consider the following cases:

Case 1: Suppose there is one list on the outer circle, say L_{001} , and two lists, say L_{011} , L_{101} on the inner circle, such that all $|L_{001}(t_1, n-1)|$, $|L_{011}(t_1, n-1)|$, and $|L_{101}(t_1, n-1)|$ are strictly positive. In this case, since L_{001} , L_{011} , and L_{101} are non-empty, it implies that L_{110} , L_{100} and L_{010} , and are empty at the beginning of super slot n . Note that there is only one uncoded transmission in this super slot, which may increase the sum of $|L_{100}(t_1, n)| + |L_{010}(t_1, n)| + |L_{110}(t_1, n)|$ by at most 1. Also note that since all coded packet transmission will either destroy coding opportunities or cause packet relocation within the set of coding opportunities involving the same user, and since $L_{100}(t_1, n)$, $L_{010}(t_1, n)$, $L_{110}(t_1, n)$ correspond to the coding opportunity involving user 3, we thus have that the sum of the lengths satisfy $RC_3(t_1, n) = |L_{100}(t_1, n)| + |L_{010}(t_1, n)| + |L_{110}(t_1, n)| \leq |L_{100}(t_1, n-1)| + |L_{010}(t_1, n-1)| + |L_{110}(t_1, n-1)| + 1$.

Case 2: Suppose there are two lists on the outer circle, say L_{100} , L_{001} , and one list from the inner circle, say L_{101} , such that all $|L_{100}(t_1, n-1)|$, $|L_{001}(t_1, n-1)|$, and $|L_{101}(t_1, n-1)|$ are strictly positive. In this case, since L_{100} , L_{001} , and L_{101} are non-empty, it implies that L_{011} , L_{110} , and L_{010} are empty at the beginning of super slot n . On the other hand, Proposition 5 implies that with probability $1 - \epsilon'_1$, either $|L_{001}(t_1, n-1)| < \epsilon'_1 \lambda B_7 / 2$ or $|L_{100}(t_1, n-1)| < \epsilon'_1 \lambda B_7 / 2$. Suppose it is $|L_{100}(t_1, n-1)|$ that is smaller than $\epsilon'_1 \lambda B_7 / 2$. We thus have $RC_3(t_1, n) \leq |L_{100}(t_1, n-1)| + |L_{010}(t_1, n-1)| + |L_{110}(t_1, n-1)| + 1 = |L_{100}(t_1, n-1)| + 1 \leq \epsilon'_1 \lambda B_7 / 2 + 1$.

Case 3: Suppose that all three lists on the outer circle, L_{100} , L_{010} , and L_{001} , satisfy that all $|L_{100}(t_1, n-1)|$, $|L_{010}(t_1, n-1)|$, and $|L_{001}(t_1, n-1)|$ are strictly positive. In this case, since L_{100} , L_{001} , and L_{010} are non-empty, it implies that L_{011} , L_{110} , and L_{101} are empty at the beginning of super slot n . On the other hand, Proposition 5 implies that with probability $1 - \epsilon'_1$, at least two $\mathbf{v}$'s in $\{100, 010, 001\}$ have $|L_{\mathbf{v}}(t_1, n-1)| \leq \frac{\epsilon'_1 \lambda B_7}{2}$. Suppose it is $|L_{100}(t_1, n-1)|$ and $|L_{010}(t_1, n-1)|$ that are smaller than $\epsilon'_1 \lambda B_7 / 2$. We thus have $RC_3(t_1, n) \leq |L_{100}(t_1, n-1)| + |L_{010}(t_1, n-1)| + |L_{110}(t_1, n-1)| + 1 = |L_{100}(t_1, n-1)| + |L_{010}(t_1, n-1)| + 1 \leq \epsilon'_1 \lambda B_7$.

Case 4: The previous cases consider the scenario that there are exactly three lists in the beginning of the super time slot n that have packets in the current time scope. In this case, we consider the scenario in which there are no more than two lists that have packets in current time scope at the beginning of one super slot, and none of them is from the outer circle. It turns out that for all such cases, the analysis is quite straightforward. For example, suppose only $|L_{110}(t_1, n-1)|$ and $|L_{101}(t_1, n-1)|$ are > 0 and all other lists have zero packets in the current time scope in the beginning of super time slot n . Then we have $RC_1(t_1, n) \leq |L_{001}(t_1, n-1)| + |L_{010}(t_1, n-1)| + |L_{011}(t_1, n-1)| + 1 = 1$. To streamline the discussion, we omit the discussion of the other subcases of Case 4.

For those cases discussed above, we have shown that that given $q(t_1) > B$, with probability $1 - \epsilon'_1$, we can find one user such that the total number of the corresponding coding opportunities in the current time scope is upper bounded by $\epsilon'_1 \lambda B_7$. Recall that $Q(t_1, n) \triangleq \min_j RC_j(t_1, n)$. As a result,

$$P(Q(t_1, n) \geq \epsilon'_1 \lambda B_7 | q(t_1) > B) \leq \epsilon'_1, \text{ for } n \geq \mathcal{N}_1.$$

Similar to the discussion in Proposition 5, the above discussion focuses on the end of super time slot $n-1$. To extend the results from the super time slot to that of the regular time slot, we again use the fact that there is only one uncoded packet transmitted by the BS in one super slot, which potentially may increase the number of coding opportunities, and since all other coded packet transmission will either destroy coding opportunities or cause packet relocation within the lists (whose packets are of the same user's coding opportunities). Therefore, for all $t > t_{\text{end}}(\mathcal{N}_1)$, we have

$$P(Q(t_1, t) \geq \epsilon'_1 \lambda B_7 + 1 | q(t_1) > B) \leq \epsilon'_1.$$

Let $B_5 = t_{\text{end}}(\mathcal{N}_1) - t_1$, and choose $B_1 = \max(B_5, B_7 + \frac{1}{\epsilon'_1 \lambda})$. Then for all $B > B_1$, we have $\epsilon'_1 > 0$, $P(Q(t_1, t_1 + B) \geq \epsilon'_1 \lambda B | q(t_1) > B) \leq \epsilon'_1$. ■

APPENDIX C PROOF FOR LEMMA 7

Proof:

Let W_n be the number of transmissions of the n -th packet when a mixed policy is used. We define Γ_n as the random variable for the transmission policy of packet n . Let $\Gamma_n = k$ be the event that we choose to apply policy π_k for packet n , and thus $\mathbf{1}_{\Gamma_n = k}$ is one if and only if we apply policy π_k to packet n . W_n can then be defined iteratively as follows:

$$W_1 = \min \left(\sum_{k=1}^M X_1^{(k)} \mathbf{1}_{\Gamma_1 = k}, \lambda \right),$$

$$\forall n = 2, \dots, N,$$

$$W_n = \min \left(\sum_{k=1}^M X_n^{(k)} \mathbf{1}_{\Gamma_n = k}, \lambda n - \sum_{i=1}^{n-1} W_i \right).$$

That is, when $\Gamma_n = k$ (when we choose policy π_k for packet n), we either continue to transmit packet n until it expires (with expiration λn) or we stop after $X_n^{(k)}$ time slots, where $X_n^{(k)}$ is the order statistic that represents the first time that exactly k

users successfully receive packet n . As a result, the number of times that the n -th packet is transmitted is the minimum of the two. We thus have the above iterative definition for W_n .

Define $S_n \triangleq \sum_{i=1}^{n-1} W_i + \sum_{k=1}^M X_n^{(k)} \mathbf{1}_{\Gamma_n=k}$ and $W_1^{n-1} \triangleq \{W_1, W_2, \dots, W_{n-1}\}$. Since $\sum_{k=1}^M X_n^{(k)} \mathbf{1}_{\Gamma_n=k}$ is independent of W_1^{n-1} , we have

$$\begin{aligned} \mathbb{E}\{e^{\tau S_n}\} &= \mathbb{E}\left\{\mathbb{E}\left\{e^{\tau S_n} \mid W_1^{n-1}\right\}\right\} \\ &= \mathbb{E}\left\{e^{(\sum_{i=1}^{n-1} W_i)\tau} \mathbb{E}\left\{e^{\tau(\sum_{k=1}^M X_n^{(k)} \mathbf{1}_{\Gamma_n=k})} \mid W_1^{n-1}\right\}\right\} \\ &= \mathbb{E}\left\{e^{\sum_{i=1}^{n-1} W_i \tau}\right\} \mathbb{E}\left\{e^{\tau(\sum_{k=1}^M X_n^{(k)} \mathbf{1}_{\Gamma_n=k})}\right\}. \end{aligned} \quad (54)$$

Also by definition $W_{n-1} \leq \sum_{k=1}^M X_{n-1}^{(k)} \mathbf{1}_{\Gamma_{n-1}=k}$, we have

$$\sum_{i=1}^{n-1} W_i \leq S_{n-1} \triangleq \sum_{i=1}^{n-2} W_i + \sum_{k=1}^M X_{n-1}^{(k)} \mathbf{1}_{\Gamma_{n-1}=k}. \quad (55)$$

By iteratively applying (54), we have

$$\mathbb{E}\{e^{\tau S_n}\} \leq \prod_{i=1}^n \mathbb{E}\left\{e^{\tau(\sum_{k=1}^M X_i^{(k)} \mathbf{1}_{\Gamma_i=k})}\right\} = \left(\mathbb{E}\left\{e^{\tau(\sum_{k=1}^M X_1^{(k)} \mathbf{1}_{\Gamma_1=k})}\right\}\right)^n,$$

where the last equality follows from that $\sum_{k=1}^M X_i^{(k)} \mathbf{1}_{\Gamma_i=k}$ has the same marginal distribution for all i . Since $\mathbb{E}\left\{\sum_{k=1}^M X_1^{(k)} \mathbf{1}_{\Gamma_1=k}\right\} = \sum_{k=1}^M \mathbb{P}(\pi_k) \mathbb{E}\{X_1^{(k)}\} = \lambda - \epsilon_1$, we can choose $\gamma \triangleq \lambda - \frac{\epsilon_1}{2}$, which is strictly larger than $\mathbb{E}\left\{\sum_{k=1}^M X_1^{(k)} \mathbf{1}_{\Gamma_1=k}\right\}$. Since $X_1^{(k)}$ is the order statistic of geometric random variables, it can be easily verified that there exists a $\delta > 0$ such that $\mathbb{E}\left\{e^{\tau(\sum_{k=1}^M X_1^{(k)} \mathbf{1}_{\Gamma_1=k})}\right\}$ exists for all τ in the open interval $(-\delta, \delta)$ containing 0. Combining with our choice of $\gamma > \mathbb{E}\left\{\sum_{k=1}^M X_1^{(k)} \mathbf{1}_{\Gamma_1=k}\right\}$, there must exist a $\tilde{\tau}_1 \in (0, \delta)$, such that

$$\mathbb{E}\left\{e^{\tilde{\tau}_1(\sum_{k=1}^M X_1^{(k)} \mathbf{1}_{\Gamma_1=k})}\right\} < e^{\tilde{\tau}_1 \gamma}.$$

We define $\text{NEBR}_n \geq k$ as the event that packet n will not expire before received by at least k users. Conditioning on that the BS chooses policy π_k , the probability that packet n can be received by at least k users is

$$\begin{aligned} \mathbb{P}(\text{NEBR}_n \geq k \mid \Gamma_n = k) &= \mathbb{P}(W_n = X_n^{(k)} \mid \Gamma_n = k) \\ &= 1 - \mathbb{P}(S_n > \lambda n \mid \Gamma_n = k) \geq 1 - \frac{1}{\mathbb{P}(\pi_k)} \mathbb{P}(S_n \geq \lambda n) \\ &\geq 1 - \frac{1}{\mathbb{P}(\pi_k)} \min_{\tau > 0} \frac{\mathbb{E}\{e^{\tau S_n}\}}{e^{\lambda n \tau}} \\ &\geq 1 - \frac{1}{\mathbb{P}(\pi_k)} \min_{\tau > 0} \frac{\mathbb{E}\left\{e^{\tau(\sum_{k=1}^M X_1^{(k)} \mathbf{1}_{\Gamma_1=k})}\right\}^n}{e^{\lambda n \tau}} \\ &\geq 1 - \frac{e^{\gamma n \tilde{\tau}_1}}{\mathbb{P}(\pi_k) e^{\lambda n \tilde{\tau}_1}} \geq 1 - \frac{e^{-\tilde{\tau}_1 \frac{\epsilon_1}{2} n}}{\mathbb{P}(\pi_k)}. \end{aligned}$$

Let NR_n be the number of users that have received at least one additional packet during the $[\sum_{i=1}^{n-1} W_i + 1, S_n]$ interval.⁷ We thus have

$$\mathbb{P}(\text{NR}_n = k \mid \Gamma_n = k) = \mathbb{P}(X_n^{(k)} < X_n^{(k+1)}) \quad (56)$$

$$\mathbb{P}(\text{NR}_n = l \mid \Gamma_n = k) = \mathbb{P}(X_n^{(l)} = X_n^{(k)}, X_n^{(l+1)} > X_n^{(k)}), \quad l = k+1, \dots, M-1, \quad (57)$$

$$\mathbb{P}(\text{NR}_n = M \mid \Gamma_n = k) = \mathbb{P}(X_n^{(M)} = X_n^{(k)}). \quad (58)$$

Namely, (56) corresponds to the case when exactly k users receive the current packet n during the $[\sum_{i=1}^{n-1} W_i + 1, S_n]$ interval;

⁷If $S_n \leq \lambda n$, then throughout the $[\sum_{i=1}^{n-1} W_i + 1, S_n]$ interval the BS always transmits packet n . However, if $S_n > \lambda n$, then other packets such as $n+1$ are also transmitted during the $[\sum_{i=1}^{n-1} W_i + 1, S_n]$ interval. The NR_n defined herein does not distinguish these two scenarios only counts the users receiving at least one additional packet within the given interval.

(57) and (58) correspond to the case when more than k users receive the current packet n during the $[\sum_{i=1}^{n-1} W_i + 1, S_n]$ interval.

Let AN_n be the actual number of users that have received packet n (without expiration) when it is transmitted. Then by the union bound, we have

$$\mathbb{P}(\text{AN}_n = l | \Gamma_n = k) = \mathbb{P}(\text{NEBR}_n \geq k, \text{NR}_n = l | \Gamma_n = k) \quad (59)$$

$$\geq \mathbb{P}(\text{NR}_n = l | \Gamma_n = k) - \frac{e^{-\tilde{\tau}_1 \frac{\epsilon_1}{2} n}}{\mathbb{P}(\pi_k)}, \quad l = k, \dots, M. \quad (60)$$

We first consider the case when the π_k policy is used. Assume that the deadline of the current packet n is at time infinity, i.e., under policy π_k , packet n will be received by at least k users before it expires. With the above assumption, the expected number of users that can receive packet n is

$$\sum_{l=k}^M l \mathbb{P}(\text{AN}_n = l | \Gamma_n = k) \quad (61)$$

$$\geq \sum_{l=k}^M l \left(\mathbb{P}(\text{NR}_n = l | \Gamma_n = k) - \frac{e^{-\tilde{\tau}_1 \frac{\epsilon_1}{2} n}}{\mathbb{P}(\pi_k)} \right) \quad (62)$$

$$\begin{aligned} &= k \mathbb{P}(X_n^{(k)} < X_n^{(k+1)}) + \sum_{l=k+1}^{M-1} l \mathbb{P}(X_n^{(l)} = X_n^{(k)}, X_n^{(l+1)} > X_n^{(k)}) \\ &\quad + M \mathbb{P}(X_n^{(M)} = X_n^{(k)}) - (M - k + 1) \frac{M + k}{2} \frac{e^{-\tilde{\tau}_1 \frac{\epsilon_1}{2} n}}{\mathbb{P}(\pi_k)} \\ &= k \mathbb{P}(X_n^{(k)} < X_n^{(k+1)}) + \sum_{l=k+1}^{M-1} k \mathbb{P}(X_n^{(l)} = X_n^{(k)}, X_n^{(l+1)} > X_n^{(k)}) \\ &\quad + k \mathbb{P}(X_n^{(M)} = X_n^{(k)}) + (M - k) \mathbb{P}(X_n^{(M)} = X_n^{(k)}) \\ &\quad + \sum_{l=k+1}^M (l - k) \mathbb{P}(X_n^{(l)} = X_n^{(k)}, X_n^{(l+1)} > X_n^{(k)}) \\ &\quad - (M - k + 1) \frac{M + k}{2} \frac{e^{-\tilde{\tau}_1 \frac{\epsilon_1}{2} n}}{\mathbb{P}(\pi_k)} \end{aligned} \quad (63)$$

$$\begin{aligned} &= k + (M - k) \mathbb{P}(X_n^{(M)} = X_n^{(k)}) \\ &\quad + \sum_{l=k+1}^{M-1} (l - k) \mathbb{P}(X_n^{(l)} = X_n^{(k)}, X_n^{(l+1)} > X_n^{(k)}) \\ &\quad - (M - k + 1) \frac{M + k}{2} \frac{e^{-\tilde{\tau}_1 \frac{\epsilon_1}{2} n}}{\mathbb{P}(\pi_k)} \end{aligned} \quad (64)$$

$$= k + \sum_{l=k+1}^M \mathbb{P}(X_n^{(l)} = X_n^{(k)}) - (M - k + 1) \frac{M + k}{2} \frac{e^{-\tilde{\tau}_1 \frac{\epsilon_1}{2} n}}{\mathbb{P}(\pi_k)}. \quad (65)$$

By plugging (60) into (61), we have (62). Then (63) follows from expansion of (62). Equation (64) and (65) follow from the total probability theorem. So the expected throughput for transmitting packet n is at least

$$\sum_{k=1}^M \mathbb{P}(\pi_k) \left(k + \sum_{l=k+1}^M \mathbb{P}(X_n^{(l)} = X_n^{(k)}) \right) - \text{Resi}(n),$$

where $\text{Resi}(n) = \sum_{k=1}^M (M - k + 1) \frac{M + k}{2} e^{-\tilde{\tau}_1 \frac{\epsilon_1}{2} n}$. Since $\mathbb{P}(X_n^{(l)} = X_n^{(k)})$ is identical for all n , we will use $X^{(l)}$ to represent

$X_n^{(l)}$ in the following for ease of exposition. Then the normalized total expected throughput is

$$\begin{aligned}
& \lim_{N \rightarrow \infty} \frac{\mathbb{E}\{N_{\text{success}}\}}{N} \\
&= \lim_{N \rightarrow \infty} \frac{\sum_{n=1}^N \sum_{k=1}^M \mathbb{P}(\pi_k) \left(k + \sum_{l=k+1}^M \mathbb{P}(X^{(l)} = X^{(k)}) \right)}{N} - \frac{\sum_{n=1}^N \text{Resi}(n)}{N} \\
&= \lim_{N \rightarrow \infty} \frac{N \sum_{k=1}^M \mathbb{P}(\pi_k) \left(k + \sum_{l=k+1}^M \mathbb{P}(X^{(l)} = X^{(k)}) \right)}{N} - \frac{\sum_{n=1}^N \text{Resi}(n)}{N} \\
&= \sum_{k=1}^M \mathbb{P}(\pi_k) \left(k + \sum_{l=k+1}^M \mathbb{P}(X^{(l)} = X^{(k)}) \right). \tag{66}
\end{aligned}$$

■

APPENDIX D PROOF FOR LEMMA 8

Proof: We will prove the lemma in two cases separately. First, we will consider Case 1.

Case 1: When $\lambda N < \frac{N}{1-(1-p)^M}$, consider any optimal solution denoted by $\bar{w}$. We will show by contradiction that any solution $\bar{w}$ not in the form of $\bar{w}_0 = \lambda N, \bar{w}_j = 0, M \geq j \geq 1$ must not be optimal. To that end, we consider the following 3 sub-cases.

Case 1.1, $\bar{w}_0 < \lambda N$, and there exists some $0 < j \leq M$, such that $\bar{w}_j > 0$. We use j' to denote the largest j such that $\bar{w}_j > 0$. We also denote the corresponding objective function value by OBJ_1 .

In this case, let $\Delta = \lambda N - \bar{w}_0$, let $\bar{w}_0^* = \bar{w}_0 + \bar{w}_{j'}$, and $\bar{w}_{j'}^* = \bar{w}_{j'} - \bar{w}_{j'}$, $\bar{w}_j^* = \bar{w}_j$, for $j > 0$ and $j \neq j'$. Namely, we reassign the values of w_0 and $w_{j'}$ to $\bar{w}_0^*$ and $\bar{w}_{j'}^*$, respectively. Note that after the reassignment, $\bar{w}^*$ still satisfy the constraints in (14)-(17). We denote the new objective function value as OBJ_2 . Since $\text{OBJ}_2 - \text{OBJ}_1 = \bar{w}_{j'}(M - (M - j'))p > 0$, the new assignment leads to a larger value. Therefore, such $\bar{w}$ cannot be optimal.

Case 1.2, $\bar{w}_0 < \lambda N$, and $\bar{w}_j = 0$, for all $1 \leq j \leq M$.

In this case, let $\bar{w}_0^* = \lambda N$, and define OBJ_1 and OBJ_2 as the objective value before and after assigning the new value of $\bar{w}_0$. Then one can easily verify that the new $\bar{w}^*$ still satisfy the constraints in (14)-(17), and $\text{OBJ}_2 > \text{OBJ}_1$. Therefore, such $\bar{w}$ cannot be optimal.

Case 1.3, $\bar{w}_0 = \lambda N$, and $\exists j'$, such that $\bar{w}_{j'} > 0$.

In this case, $\bar{w}_0 + \bar{w}_{j'} > \lambda N$, which contradicts with the constraint (14). Such $\bar{w}$ is not a feasible solution.

From the above discussion, the proof for the Case 1 is complete.

Case 2: When $\lambda N \geq \frac{N}{1-(1-p)^M}$, consider any optimal solution denoted by $\bar{w}$. We will also show by contradiction that if the solution is not the in the form of (22), then it is not optimal. To that end, we consider the following 3 sub-cases.

Case 2.1, $\bar{w}_0 < \frac{N}{1-(1-p)^M}$, and there exists some $0 < j \leq M$, such that $\bar{w}_j > 0$. Again, we use j' to denote the largest j satisfying $\bar{w}_j > 0$.

In this case, let $\Delta = \frac{N}{1-(1-p)^M} - \bar{w}_0$, let $\bar{w}_0^* = \bar{w}_0 + \bar{w}_{j'}$, and $\bar{w}_{j'}^* = \bar{w}_{j'} - \bar{w}_{j'}$, $\bar{w}_j^* = \bar{w}_j$, for $j > 0$ and $j \neq j'$. Namely, we reassign the values of w_0 and $w_{j'}$ to $\bar{w}_0^*$ and $\bar{w}_{j'}^*$, respectively. Note that after the reassignment, $\bar{w}^*$ still satisfy the constraints in (14)-(17). We denote the new objective function value as OBJ_2 . Since $\text{OBJ}_2 - \text{OBJ}_1 = \bar{w}_{j'}(M - (M - j'))p > 0$, therefore, such $\bar{w}$ cannot be optimal.

Case 2.2 $\bar{w}_0 < \frac{N}{1-(1-p)^M}$, and $\forall 0 < j' \leq M, \bar{w}_{j'} = 0$.

In this case, we can set the value of $\bar{w}_0^* = \frac{N}{1-(1-p)^M}$, and define OBJ_1 and OBJ_2 as the objective value before and after assigning the new value of $\bar{w}_0$. Then one can easily verify that the new $\bar{w}^*$ still satisfy the constraints in (14)-(17), and $\text{OBJ}_2 > \text{OBJ}_1$. Therefore, such $\bar{w}$ cannot be optimal.

Case 2.3, $\bar{w}_0 = \frac{N}{1-(1-p)^M}$ and there exists a j' , such that

$$\begin{aligned}
\bar{w}_0 &= \frac{N}{1 - (1-p)^M} \\
\bar{w}_j(1 - (1-p)^{M-j}) &= \sum_{s=0}^{j-1} \bar{w}_s \binom{M-s}{j-s} p^{j-s} (1-p)^{M-j} \text{ for } j \leq j', \\
\bar{w}_{j'+1}(1 - (1-p)^{M-j'-1}) &< \sum_{s=0}^{j'} \bar{w}_s \binom{M-s}{j-s} p^{j'+1-s} (1-p)^{M-j'-1} \\
&\exists j \in (j' + 1, M], \text{ such that } \bar{w}_j > 0. \tag{67}
\end{aligned}$$

Among all $j \in (j' + 1, M]$ satisfying (67), we use j'' to denote the largest such j value.

Then let $\Delta = \frac{\sum_{s=0}^{j'} \bar{w}_s \binom{M-s}{j'+1-s} p^{j'+1-s} (1-p)^{M-j'-1}}{1-(1-p)^{M-j'-1}} - \bar{w}_{j'+1}$, assign $\bar{w}_{j'+1}^* = \bar{w}_{j'+1} + \bar{w}_{j''}$, $\bar{w}_{j''}^* = \bar{w}_{j''} - \bar{w}_{j''}$, $\bar{w}_j^* = \bar{w}_j$, for $j > j' + 1$ and $j \neq j''$.

It can be easily verified that after the reassignment of $\bar{w}_{j'+1}^*$ and $\bar{w}_{j''}^*$, they still satisfy the constraints in (14)-(17), and also the difference of the objective function value becomes $\text{OBJ}_2 - \text{OBJ}_1 = \bar{w}_{j''}(M - j' - 1 - (M - j''))p > 0$. Therefore, such $\bar{w}$ cannot be optimal.

The above discussion implies the optimal solution for Case 2 is

$$\begin{aligned} \bar{w}_0 &< \frac{N}{1 - (1-p)^M} \\ \exists 1 < m \leq M-1, \text{ such that} \\ 0 &< \bar{w}_j(1 - (1-p)^{M-j}) = \sum_{s=0}^{j-1} \bar{w}_s \binom{M-s}{j-s} p^{j-s} (1-p)^{M-j} \text{ for } j \leq m, \\ 0 &< \bar{w}_j(1 - (1-p)^{M-j}) < \sum_{s=0}^{j-1} \bar{w}_s \binom{M-s}{j-s} p^{j-s} (1-p)^{M-j} \text{ for } j = m+1, \\ \bar{w}_j &= 0, \text{ for } j > m+1. \end{aligned}$$

■

APPENDIX E PROOF FOR LEMMA 9

Proof:

$$\text{Since } \bar{w}_j = \mathbb{E} \left\{ X^{(j+1)} - X^{(j)} \right\} N \sum_{k=j+1}^M \mathbb{P}(\pi_k) = 0, \forall j > m,$$

$$\text{then, } \forall j > m, \text{ we have } \sum_{k=j+1}^M \mathbb{P}(\pi_k) = 0,$$

from which we can easily deduce $\mathbb{P}(\pi_k) = 0, \forall k > m+1$.

■

APPENDIX F PROOF FOR PROPOSITION 10

Proof: We first consider the case when $\lambda N < \frac{N}{1-(1-p)^M}$.

Since the optimal solution is $\bar{w}_0 = \lambda N, \bar{w}_j = 0, 1 \leq j \leq M$, (24) implies $\mathbb{P}(\pi_k) = 0, 1 < k \leq M$. As we already know $\bar{w}_0 = \lambda N$, then by (23), we can easily derive $\mathbb{P}(\pi_1) = \lambda(1 - (1-p)^M)$. (Here we actually have $\mathbb{P}(\pi_0) = 1 - \lambda(1 - (1-p)^M)$.)

Now we will discuss the case when $\lambda N \geq \frac{N}{1-(1-p)^M}$. As the first step to approach this problem, we will calculate $\mathbb{P}(X^{(j+1)} > X^{(j)})$.

For any integer $0 < i < M$, consider any strictly non-negative integer sequence $m_1, m_2, \dots, m_i$, such that $\sum_{b=1}^i m_b = i$. $\exists l_1, 0 < l_1 \leq M$, such that $m_b > 0$, for all $b \leq l_1$, and $m_b = 0$, for all $l_1 < b \leq i$. Let $m_0 = 0$. We use $\mathcal{M}_i$ to denote the set of such sequences.

Given any such sequence above, we define for any $0 < l \leq l_1$,

$$f_l(m_1, \dots, m_l) = \frac{\binom{M - \sum_{b=0}^{l-1} m_b}{m_l} p^{m_l} (1-p)^{M - \sum_{b=1}^l m_b}}{1 - (1-p)^{M - \sum_{b=1}^{l-1} m_b}}, \quad (68)$$

$$\text{for } 0 < m_1, \dots, m_l < i < M. \quad (69)$$

Given any integer $l > 0$, define

$$\Phi_h^l = \left\{ X^{(l+1)} > X^{(l)} = \dots = X^{(h+1)} > X^{(h)} \right\},$$

for $h = 1, 2, \dots, l-1$, then for any given non-negative integer sequence $m_1, m_2, \dots, m_i$, such that $\sum_{b=1}^{l_1} m_b = i$, we are going to prove

$$\mathbb{P} \left(\bigcap_{l=1}^{l_1} \Phi_{\sum_{b=0}^{l-1} m_b}^{\sum_{b=1}^l m_b} \right) = \prod_{b=1}^{l_1} f_b(m_1, \dots, m_b), \quad (70)$$

This probability is that $X^{(1)}$ to $X^{(i)}$, $(X^{(i)} < X^{(i+1)})$, takes l_1 different values, and for each value there are $m_1, \dots, m_{l_1}$ candidates among $X^{(1)}, \dots, X^{(i)}$ equal to it respectively, starting from the smallest to the largest.

As an intermediate step to calculate $P(X^{(i+1)} > X^{(i)})$, we first sketch the proof for (70) when $i = 1$, $m_1 = i$. Recall that we temporarily define the time at which the BS decides to transmit packet n for the first time as the new origin.

$$\begin{aligned}
P(\Phi_0^{m_1}) &= \sum_{k_1=1}^{\infty} P(X^{(1)} = k_1) P(X^{(m_1+1)} > X^{(m_1)} = X^{(1)} | X^{(1)} = k_1) \\
&= \sum_{k_1=1}^{\infty} P(X^{(1)} = k_1) \frac{P(X^{(m_1+1)} > X^{(m_1)} = X^{(1)}, X^{(1)} = k_1)}{P(X^{(1)} = k_1)} \\
&= \sum_{k_1=1}^{\infty} P(X^{(1)} = k_1) \frac{((1-p)^M)^{k_1-1} \binom{M}{m_1} p^{m_1} (1-p)^{M-m_1}}{((1-p)^M)^{k_1-1} (1 - (1-p)^M)} \\
&= \sum_{k_1=1}^{\infty} P(X^{(1)} = k_1) \frac{\binom{M}{m_1} p^{m_1} (1-p)^{M-m_1}}{1 - (1-p)^M} \\
&= \frac{\binom{M}{m_1} p^{m_1} (1-p)^{M-m_1}}{1 - (1-p)^M},
\end{aligned} \tag{71}$$

where (71) can be explained as: $((1-p)^M)^{k_1-1}$ is the probability that none of all users have received any packet in $k_1 - 1$ time slots; $\binom{M}{m_1} p^{m_1} (1-p)^{M-m_1}$ is the probability that exactly m_1 users receive one packet k_1 slots after the new origin; $1 - (1-p)^M$ is the probability that at least one user receives one packet k_1 slots after the new origin. So the proof for (70) when $i = 1$ is complete. We also sketch the proof for $i = 2$.

$$\begin{aligned}
P(\Phi_0^{m_1} \cap \Phi_{m_1}^{m_1+m_2}) &= \sum_{k_1=1}^{\infty} P(X^{(1)} = k_1) \frac{((1-p)^M)^{k_1-1} \binom{M}{m_1} p^{m_1} (1-p)^{M-m_1}}{((1-p)^M)^{k_1-1} (1 - (1-p)^M)} \\
&\quad \sum_{k_2=1}^{\infty} P(X^{(m_1+1)} = k_1 + k_2) \\
&\quad \frac{((1-p)^{M-m_1})^{k_2-1} \binom{M-m_1}{m_2} p^{m_2} (1-p)^{M-m_1-m_2}}{((1-p)^{M-m_1})^{k_2-1} (1 - (1-p)^{M-m_1})} \\
&= \sum_{k_1=1}^{\infty} P(X^{(1)} = k_1) \frac{\binom{M}{m_1} p^{m_1} (1-p)^{M-m_1}}{1 - (1-p)^M} \\
&\quad \sum_{k_2=1}^{\infty} P(X^{(m_1+1)} = k_1 + k_2) \frac{\binom{M-m_1}{m_2} p^{m_2} (1-p)^{M-m_1-m_2}}{1 - (1-p)^{M-m_1}} \\
&= \frac{\binom{M}{m_1} p^{m_1} (1-p)^{M-m_1}}{1 - (1-p)^M} \frac{\binom{M-m_1}{m_2} p^{m_2} (1-p)^{M-m_1-m_2}}{1 - (1-p)^{M-m_1}},
\end{aligned} \tag{72}$$

$$= \sum_{k_1=1}^{\infty} P(X^{(1)} = k_1) \frac{\binom{M}{m_1} p^{m_1} (1-p)^{M-m_1}}{1 - (1-p)^M} \tag{73}$$

where (72) can be explained as: $((1-p)^{M-m_1})^{k_2-1}$ is the probability that for k_2 time slots after the time when m_1 users have received their packets, none of the other users has received anything; $\binom{M-m_1}{m_2} p^{m_2} (1-p)^{M-m_1-m_2}$ is the probability that among those $M - m_1$ users, m_2 users receive one packet simultaneously; $(1 - (1-p)^{M-m_1})$ is the probability that at

least one user receives one packet $k_1 + k_2$ slots after the new origin. By the similar arguments, we can have

$$\begin{aligned}
& \mathbb{P}(\cap_{l=1}^{l_1} \Phi_{\sum_{b=0}^{l-1} m_b}^{\sum_{b=1}^l m_b}) \\
&= \frac{\binom{M}{m_1} p^{m_1} (1-p)^{M-m_1}}{1 - (1-p)^M} \cdots \\
&\quad \cdots \frac{\binom{M - \sum_{n=1}^{l_1-1} m_n}{m_{l_1}} p^{m_{l_1}} (1-p)^{M - \sum_{n=1}^{l_1-1} m_n}}{1 - (1-p)^{M - \sum_{n=1}^{l_1-1} m_n}} \\
&= \prod_{b=1}^{l_1} f_b(m_1, \dots, m_b),
\end{aligned}$$

which implies that

$$\mathbb{P}(\cap_{l=1}^i \Phi_{\sum_{b=0}^{l-1} m_b}^{\sum_{b=1}^l m_b}) = \prod_{b=1}^i f_b(m_1, \dots, m_b).$$

Since

$$\{X^{(i+1)} > X^{(i)}\} = \cup_{\{m_b\}_{0 \leq b \leq i} \in \mathcal{M}_i} \cap_{l=1}^i \Phi_{\sum_{b=0}^{l-1} m_b}^{\sum_{b=1}^l m_b},$$

We can easily obtain

$$\begin{aligned}
& \mathbb{P}(X^{(i+1)} > X^{(i)}) \\
&= \sum_{\{m_b\}_{0 \leq b \leq i} \in \mathcal{M}_i} \prod_{b=1}^i f_b(m_1, \dots, m_b).
\end{aligned} \tag{74}$$

After calculating $\mathbb{P}(X^{(i+1)} > X^{(i)})$, now we are going to show that, when

$$\begin{aligned}
\bar{w}_0 &= \frac{N}{1 - (1-p)^M} \\
0 < \bar{w}_j (1 - (1-p)^{M-j}) &= \sum_{s=0}^{j-1} \bar{w}_s \binom{M-s}{j-s} p^{j-s} (1-p)^{M-j},
\end{aligned} \tag{75}$$

$$\begin{aligned}
& \text{for } j = 1, 2, \dots, m \\
\bar{w}_j &= 0, \quad \text{for } j > m,
\end{aligned} \tag{76}$$

$\bar{w}_j = \mathbb{E}\{X^{(j+1)} - X^{(j)}\}(1 - (1-p)^M) \bar{w}_0$ for all $j < m$, and $\mathbb{P}(\pi_{m+1}) = 1$, $\mathbb{P}(\pi_k) = 0$, for $k \neq m+1$. We need to note that (76) is a special case of (22).

Now we will prove

$$\bar{w}_j = \mathbb{E}\{X^{(j+1)} - X^{(j)}\}(1 - (1-p)^M) \bar{w}_0. \tag{77}$$

Among them,

$$\begin{aligned}
& \mathbb{E}\{X^{(j+1)} - X^{(j)}\} \\
&= \mathbb{P}(X^{(j+1)} > X^{(j)}) \mathbb{E}\{X^{(j+1)} - X^{(j)} | X^{(j+1)} > X^{(j)}\} \\
&= \frac{1}{1 - (1-p)^{M-j}} \mathbb{P}(X^{(j+1)} > X^{(j)}).
\end{aligned} \tag{78}$$

We will prove (77) by induction.

When $j = 0$, it is satisfied because

$$\mathbb{E}\{X^{(1)} - 0\} = \frac{1}{1 - (1-p)^M} \mathbb{P}(X^{(1)} > 0) = \frac{1}{1 - (1-p)^M}.$$

Suppose when $j = i$, $1 < i < m$,

$$\bar{w}_i = \mathbb{E}\{X^{(i+1)} - X^{(i)}\}(1 - (1-p)^M) \bar{w}_0, \tag{79}$$

Then when $j = i + 1$, using (76), we have

$$\bar{w}_{i+1} = \frac{\sum_{s=0}^i \bar{w}_s \binom{M-s}{i+1-s} p^{i+1-s} (1-p)^{M-i-1}}{1 - (1-p)^{M-i-1}}$$

After applying (79), we obtain

$$\bar{w}_{i+1} = \frac{\sum_{s=0}^i \bar{w}_0 (1 - (1-p)^M) \mathbb{E}\{X^{(s+1)} - X^{(s)}\} \binom{M-s}{i+1-s} p^{i+1-s} (1-p)^{M-i-1}}{1 - (1-p)^{M-i-1}}.$$

Because of (78),

$$\bar{w}_{i+1} = \frac{\bar{w}_0 (1 - (1-p)^M) \sum_{s=0}^i \frac{1}{1 - (1-p)^{M-s}} \mathbb{P}(X^{(s+1)} - X^{(s)}) \binom{M-s}{i+1-s} p^{i+1-s} (1-p)^{M-i-1}}{1 - (1-p)^{M-i-1}}.$$

Since we have (74)

$$\begin{aligned} \bar{w}_{i+1} &= \frac{\bar{w}_0 (1 - (1-p)^M) \sum_{s=0}^i \sum_{\{m_l\}_{0 \leq l \leq s} \in \mathcal{M}_s} \prod_{b=1}^s f_b(m_1, \dots, m_b) \binom{M-s}{i+1-s} p^{i+1-s} (1-p)^{M-i-1}}{1 - (1-p)^{M-i-1} \frac{1}{1 - (1-p)^{M-s}}} \end{aligned}$$

Since

$$\begin{aligned} & \sum_{s=0}^i \sum_{\{m_l\}_{0 \leq l \leq s} \in \mathcal{M}_s} \prod_{b=1}^s f_b(m_1, \dots, m_b) \frac{\binom{M-s}{i+1-s} p^{i+1-s} (1-p)^{M-i-1}}{1 - (1-p)^{M-s}} \\ &= \sum_{\{m_l\}_{0 \leq l \leq i+1} \in \mathcal{M}_{i+1}} \prod_{b=1}^{i+1} f_b(m_1, \dots, m_b), \\ \bar{w}_{i+1} &= \frac{\bar{w}_0 (1 - (1-p)^M) \sum_{\{m_l\}_{0 \leq l \leq i+1} \in \mathcal{M}_{i+1}} \prod_{b=1}^{i+1} f_b(m_1, \dots, m_b)}{1 - (1-p)^{M-i-1}} \\ &= \bar{w}_0 (1 - (1-p)^M) \frac{1}{1 - (1-p)^{M-i-1}} \mathbb{P}(X^{(i+2)} > X^{(i+1)}) \\ &= \mathbb{E}\{X^{(i+2)} - X^{(i+1)} | X^{(i+2)} > X^{(i+1)}\} \mathbb{P}(X^{(i+2)} > X^{(i+1)}) (1 - (1-p)^M) \bar{w}_0 \\ &= \mathbb{E}\{X^{(i+2)} - X^{(i+1)}\} (1 - (1-p)^M) \bar{w}_0. \end{aligned}$$

By the induction, we can conclude $\bar{w}_j = \mathbb{E}\{X^{(j+1)} - X^{(j)}\} \bar{w}_0 (1 - (1-p)^M)$, $\forall j < m$. Since $\bar{w}_0 = \frac{N}{1 - (1-p)^M}$, we can also say $\frac{\bar{w}_j}{N \mathbb{E}\{X^{(j+1)} - X^{(j)}\}} = 1$. Therefore, (23) and (24) implies $\mathbb{P}(\pi_{m+1}) = 1$, given (76).

After showing $\mathbb{P}(\pi)$ for the special case (76), now we are going to prove the general case, when the solution to (13) is in the form of (22).

Since

$$\bar{w}_j = \mathbb{E}\{X^{(j+1)} - X^{(j)}\} (1 - (1-p)^M) \bar{w}_0, 1 \leq j < m,$$

$\bar{w}_j$, $1 < j < m$ has a linear relationship with $\bar{w}_0$.

Assume for $j = 0, \dots, M-1$, $\bar{w}_j = \bar{w}'_j + \bar{w}''_j$, for $k = 1, \dots, M$, $\mathbb{P}(\pi_k) = \mathbb{P}(\pi_k)' + \mathbb{P}(\pi_k)''$. Note that for all j , $\bar{w}'_j \geq 0$, $\bar{w}''_j \geq 0$, and for all k , $\mathbb{P}(\pi_k)' \geq 0$, $\mathbb{P}(\pi_k)'' \geq 0$. In addition, the following equalities are satisfied.

$$\begin{aligned} \bar{w}'_0 &= \mathbb{E}\{X^{(1)}\} N \sum_{k=1}^M \mathbb{P}(\pi_k)' \\ \bar{w}'_j &= \mathbb{E}\{X^{(j+1)} - X^{(j)}\} N \sum_{k=j+1}^M \mathbb{P}(\pi_k)', j = 1, 2, \dots, M-1. \end{aligned} \tag{80}$$

$$\begin{aligned} \bar{w}''_0 &= \mathbb{E}\{X^{(1)}\} N \sum_{k=1}^M \mathbb{P}(\pi_k)'' \\ \bar{w}''_j &= \mathbb{E}\{X^{(j+1)} - X^{(j)}\} N \sum_{k=j+1}^M \mathbb{P}(\pi_k)'', j = 1, 2, \dots, M-1. \end{aligned} \tag{81}$$

Let

$$\begin{aligned}\bar{w}'_0 &= \frac{\bar{w}_m(1 - (1 - p)^{M-m})}{\sum_{s=0}^{m-1} \bar{w}_s \binom{M-s}{m-s} p^{m-s} (1 - p)^{M-m}} \frac{N}{1 - (1 - p)^M} \\ \bar{w}'_j &= \mathbb{E}\{X^{(j+1)} - X^{(j)}\}(1 - (1 - p)^M) \bar{w}'_0, \quad 1 \leq j < m, \\ \bar{w}'_j &= 0, j > m.\end{aligned}$$

In particular,

$$\bar{w}'_m = \mathbb{E}\{X^{(m+1)} - X^{(m)}\}(1 - (1 - p)^M) \bar{w}'_0. \quad (82)$$

Let

$$\begin{aligned}\bar{w}''_0 &= \left(1 - \frac{\bar{w}_m(1 - (1 - p)^{M-m})}{\sum_{s=0}^{m-1} \bar{w}_s \binom{M-s}{m-s} p^{m-s} (1 - p)^{M-m}}\right) \frac{N}{1 - (1 - p)^M} \\ \bar{w}''_j &= \mathbb{E}\{X^{(j+1)} - X^{(j)}\}(1 - (1 - p)^M) \bar{w}''_0, \quad 1 \leq j < m, \\ \bar{w}''_j &= 0, j \geq m.\end{aligned}$$

By our construction,

$$\bar{w}_m = \bar{w}'_m = \mathbb{E}\{X^{(m+1)} - X^{(m)}\}(1 - (1 - p)^M) \bar{w}'_0,$$

and

$$\begin{aligned}\bar{w}'_j &= \mathbb{E}\{X^{(j+1)} - X^{(j)}\}N \sum_{k=j+1}^M \mathbb{P}(\pi_k)' = 0, \forall j > m, \\ \text{which means } \mathbb{P}(\pi_k)' &= 0, k > m + 1.\end{aligned}$$

Also,

$$\begin{aligned}\bar{w}''_j &= \mathbb{E}\{X^{(j+1)} - X^{(j)}\}N \sum_{k=j+1}^M \mathbb{P}(\pi_k)'' = 0, \forall j \geq m, \\ \text{which means } \mathbb{P}(\pi_k)'' &= 0, k > m\end{aligned}$$

$\mathbb{P}(\pi_k) = \mathbb{P}(\pi_k)' + \mathbb{P}(\pi_k)''$ implies $\mathbb{P}(\pi_k) = 0$, for $k > m + 1$.

We can refer to the special case (76) proven above, thus by (82) and (80), we have

$$\begin{aligned}\mathbb{P}(\pi_k)' &= 0, \quad k \leq m \\ \mathbb{P}(\pi_{m+1}') &= \frac{\bar{w}_m(1 - (1 - p)^{M-m})}{\sum_{s=0}^{m-1} \bar{w}_s \binom{M-s}{m-s} p^{m-s} (1 - p)^{M-m}}.\end{aligned}$$

Similarly, we have

$$\begin{aligned}\mathbb{P}(\pi_k'') &= 0, \quad k < m \\ \mathbb{P}(\pi_m'') &= 1 - \frac{\bar{w}_m(1 - (1 - p)^{M-m})}{\sum_{s=0}^{m-1} \bar{w}_s \binom{M-s}{m-s} p^{m-s} (1 - p)^{M-m}}.\end{aligned}$$

So

$$\begin{aligned}\mathbb{P}(\pi_{m+1}) &= \frac{\bar{w}_m(1 - (1 - p)^{M-m})}{\sum_{s=0}^{m-1} \bar{w}_s \binom{M-s}{m-s} p^{m-s} (1 - p)^{M-m}} \\ \mathbb{P}(\pi_m) &= 1 - \frac{\bar{w}_m(1 - (1 - p)^{M-m})}{\sum_{s=0}^{m-1} \bar{w}_s \binom{M-s}{m-s} p^{m-s} (1 - p)^{M-m}}.\end{aligned}$$

In conclusion, for the general case (22), we can obtain $P(\pi)$ by solving (23) and (24).

$$\begin{aligned} P(\pi_k) &= 0, \quad k < m, \text{ or } k > m+1, \\ P(\pi_m) &= 1 - \frac{\bar{w}_m(1 - (1-p)^{M-m})}{\sum_{s=0}^{m-1} \bar{w}_s \binom{M-s}{m-s} p^{m-s} (1-p)^{M-m}}, \\ P(\pi_{m+1}) &= \frac{\bar{w}_m(1 - (1-p)^{M-m})}{\sum_{s=0}^{m-1} \bar{w}_s \binom{M-s}{m-s} p^{m-s} (1-p)^{M-m}}. \end{aligned}$$

Especially, when the special case (76) occurs, we have $P(\pi_{m+1}) = 1$, and $P(\pi_k) = 0, \forall k \neq m+1$.

Also it's very easy to see that there are only at most two $P(\pi_k) > 0, 1 \leq k \leq M$, when $M > 2$. ■

APPENDIX G PROOF FOR THEOREM 11

Proof: When we characterize the upper bound of the uncoded case, the normalized expected throughput for UBLP is $\frac{\sum_{j=0}^{M-1} (M-j)p\bar{w}_j}{N}$. Since (23) and (24) indicate the relationships between $\bar{w}$ and $P(\pi)$, the throughput for UBLP can be expressed as

$$\frac{\sum_{j=0}^M (M-j)p\bar{w}_j}{N} \tag{83}$$

$$\begin{aligned} &= \frac{\sum_{j=0}^{M-1} (M-j)p \sum_{k=j+1}^M P(\pi_k) N E\{X^{(j+1)} - X^{(j)}\}}{N} \\ &= \sum_{k=1}^M P(\pi_k) p \sum_{j=0}^{k-1} (M-j) E\{X^{(j+1)} - X^{(j)}\}. \end{aligned} \tag{84}$$

Next we are going to show that $p \sum_{j=0}^{k-1} (M-j) E\{X^{(j+1)} - X^{(j)}\} = k + \sum_{j=k+1}^M P(X^{(j)} = X^{(k)})$. To show this, we first need to prove $\sum_{j=k+1}^M P(X^{(j)} > X^{(k)}) = p \sum_{j=k+1}^M E\{X^{(j)} - X^{(k)}\}$.

Recall that $\Phi_h^l = \{X^{(l+1)} > X^{(l)} = \dots = X^{(h+1)} > X^{(h)}\}$ for $h < l$. For each $1 \leq k < M$, we use $\Phi_{k,k+i}$ to denote $\{X^{(k+i)} > X^{(k+i-1)} = X^{(k)}\}$, $0 < i \leq M-k$. It follows that, for $j \in [k+1, M]$, $\{X^{(j)} > X^{(k)}\} = \cup_{i=1}^{j-k} \Phi_{k,k+i}$.

Since for any integer $i \in [1, j-k]$, by definition of $\Phi_{k,k+i}$, we have

$$E\{X^{(j)} - X^{(k)} \mid \Phi_{k,k+i}\} = E\{X^{(j)} - X^{(k+i-1)} \mid \Phi_{k,k+i}\},$$

which implies

$$\begin{aligned} E\{X^{(j)} - X^{(k)}\} &= \sum_{i=1}^{j-k} P(\Phi_{k,k+i}) E\{X^{(j)} - X^{(k)} \mid \Phi_{k,k+i}\} \\ &= \sum_{i=1}^{j-k} P(\Phi_{k,k+i}) E\{X^{(j)} - X^{(k+i-1)} \mid \Phi_{k,k+i}\}. \end{aligned}$$

Thus, we have

$$\begin{aligned} &\sum_{j=k+1}^M E\{X^{(j)} - X^{(k)}\} \\ &= \sum_{j=k+1}^M \sum_{i=1}^{j-k} P(\Phi_{k,k+i}) E\{X^{(j)} - X^{(k+i-1)} \mid \Phi_{k,k+i}\} \\ &= \sum_{i=1}^{M-k} P(\Phi_{k,k+i}) \sum_{l=k+i}^M E\{X^{(l)} - X^{(k+i-1)} \mid \Phi_{k,k+i}\}. \end{aligned} \tag{85}$$

Recall that for any M i.i.d. geometrically distributed random variables $X^1, X^2, \dots, X^M$ with success probability p , $X^{(k)}$ is the k -th order statistics, and we have $X^{(1)} \leq X^{(2)} \leq \dots \leq X^{(M)}$. Conditioning on $\Phi_{k,k+i}$, for any j , such that $X^j > X^{(k+i-1)}$, the conditional distribution of $X^j - X^{(k+i-1)}$ is still geometric, with success probability p . Let $\Theta = \{j : X^j > X^{(k+i-1)}\}$. Conditioning on $\Phi_{k,k+i}$, $|\Theta| = M-k-i+1$. We notice that the conditional distribution of $\sum_{m=k+i}^M (X^{(m)} - X^{(k+i-1)})$ is the same as that of $\sum_{m \in \Theta} (X^m - X^{(k+i-1)})$. It follows that $\sum_{l=k+i}^M E\{X^{(l)} - X^{(k+i-1)} \mid \Phi_{k,k+i}\} = \sum_{j \in \Theta} E\{X^j - X^{(k+i-1)} \mid \Phi_{k,k+i}\} = \frac{M-k-i+1}{p}$. Then by (85), we have

$$\sum_{j=k+1}^M \mathbb{E} \{X^{(j)} - X^{(k)}\} = \sum_{i=1}^{M-k} \mathbb{P}(\Phi_{k,k+i}) \frac{M-k-i+1}{p}. \quad (86)$$

Since $\{X^{(j)} > X^{(k)}\} = \cup_{i=1}^{j-k} \Phi_{k,k+i}$, and $\Phi_{k,k+i}$, $i = 1, \dots, j-k$, are disjoint, then for any j, k , such that $0 < k < j \leq M$, we have

$$\mathbb{P}(X^{(j)} > X^{(k)}) = \sum_{i=1}^{j-k} \mathbb{P}(\Phi_{k,k+i}).$$

Therefore,

$$\sum_{j=k+1}^M \mathbb{P}(X^{(j)} > X^{(k)}) = \sum_{j=k+1}^M \sum_{i=1}^{j-k} \mathbb{P}(\Phi_{k,k+i}) \quad (87)$$

$$= \sum_{i=1}^{M-k} (M-k-i+1) \mathbb{P}(\Phi_{k,k+i}). \quad (88)$$

Thus, using the equality in (86) and (88) yields:

$$\sum_{j=k+1}^M \mathbb{P}(X^{(j)} > X^{(k)}) = p \sum_{j=k+1}^M \mathbb{E} \{X^{(j)} - X^{(k)}\}. \quad (89)$$

Since for any j, k , such that $0 < k < j \leq M$, we have $\mathbb{P}(X^{(j)} = X^{(k)}) = 1 - \mathbb{P}(X^{(j)} > X^{(k)})$, and $p \sum_{j=1}^M \mathbb{E} \{X^{(j)}\} = p \sum_{j=1}^M \mathbb{E} \{X^j\} = M$, we plug these two equalities into (89), and derive

$$\begin{aligned} & \sum_{j=k+1}^M (1 - \mathbb{P}(X^{(j)} = X^{(k)})) \\ &= p \sum_{j=k+1}^M \mathbb{E} \{X^{(j)}\} - p \sum_{j=k+1}^M \mathbb{E} \{X^{(k)}\} \\ &= p \sum_{j=1}^M \mathbb{E} \{X^{(j)}\} - p \sum_{j=1}^{k-1} \mathbb{E} \{X^{(j)}\} - p(M-k+1) \mathbb{E} \{X^{(k)}\}. \end{aligned}$$

So

$$\begin{aligned} & M - k - \sum_{j=k+1}^M \mathbb{P}(X^{(j)} = X^{(k)}) \\ &= M - p \sum_{j=1}^{k-1} \mathbb{E} \{X^{(j)}\} - p(M-k+1) \mathbb{E} \{X^{(k)}\} \\ &= M - p \sum_{j=0}^{k-1} (M-j) \mathbb{E} \{X^{(j+1)} - X^{(j)}\}. \end{aligned}$$

This implies that

$$k + \sum_{j=k+1}^M \mathbb{P}(X^{(j)} = X^{(k)}) = p \sum_{j=0}^{k-1} (M-j) \mathbb{E} \{X^{(j+1)} - X^{(j)}\}. \quad (90)$$

Therefore, by combining (90) and (84) we can conclude that

$$\frac{\sum_{j=0}^M (M-j) p \bar{w}_j}{N} = \sum_{k=1}^M \mathbb{P}(\pi_k) \left(k + \sum_{j=k+1}^M \mathbb{P}(X^{(j)} = X^{(k)}) \right).$$

■

REFERENCES

- [1] "Purdue stadium," <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, February 2003.
- [4] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, November 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, October 2006.
- [6] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. of Allerton Conference*, 2003.
- [7] J.-K. Sundararajan, D. Shah, and M. Médard, "ARQ for network coding," in *Proc. of ISIT*, 2008.
- [8] E. Erez, M. Effros, and T. Ho, "Network codes with deadlines," in *Proc. of Allerton Conference*, 2008.
- [9] A. Eryilmaz, A. Ozdaglar, and M. Médard, "On delay performance gains from network coding," in *Proc. of CISS*, 2006.
- [10] J.-K. Sundararajan, D. Shah, and M. Médard, "Online network coding for optimal throughput and delay – the three-receiver case," in *Arxiv preprint arXiv*, 2008.
- [11] J.-K. Sundararajan, P. Sadeghi, and M. Médard, "A feedback-based adaptive broadcast coding scheme for reducing in-order delivery delay," in *NetCod*, 2009.
- [12] W. Yeow, A. Hoang, and C. Tham, "Minimizing delay for multicast-streaming in wireless networks with network coding," in *Proc. of INFOCOM*, 2009.
- [13] E. Drinea, C. Fragouli, and L. Keller, "Delay with network coding and feedback," in *Proc. of ISIT*, 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] —, "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/>, 2009.