Throughput and Delay Analysis on Uncoded and Coded Wireless Broadcast with Hard Deadline Constraints

Xiaohang Li, Chih-Chun Wang, and Xiaojun Lin
Center for Wireless Systems and Applications, School of ECE, Purdue University, West Lafayette, IN 47906
Email: {li179, chihw}@purdue.edu, linx@ecn.purdue.edu

***Abstract*—Multimedia streaming applications have stringent QoS requirements. Typically each packet is associated with a packet delivery deadline. This work models and considers real-time streaming broadcast over the downlink of a single cell. The broadcast capacity of the system subject to deadline constraints are studied for both uncoded and coded wireless broadcast schemes. For the uncoded scenario, an optimal transmission policy is devised based on finite-horizon dynamic programming, and a closed-form expression of the optimal throughput is developed in the asymptotic regime, as the size of the file approaches infinity. For the coded scenario, the optimal capacity in the asymptotic regime of large file size is also derived, which is strictly larger than that of the non-coding policies. A simple transmission policy is proposed that can achieve the asymptotic capacity while maintaining finite transmission delay (queueing + decoding dleay). A queue-length-based Lyapunov function is used to show the optimality of this policy. Simulation shows that the simple coding policy outperforms the best non-coding policies even for broadcasting files of small sizes.**

I. Introduction

The advance of the broadband wireless technologies has triggered exponential growth of the number of new services provided over cellular networks. Among them, wireless video streaming for multiple receivers has drawn substantial interests among the networking community. In wireless streaming, a large file is transmitted to multiple users through the wireless downlink. The stringent requirements on the quality of services (QoS) in video streaming pose a *delivery deadline constraint* for each packet, i.e., each video packet *expires* after a predefined deadline and is then considered useless for any receivers. In addition, the random and unreliable wireless channel poses another challenge for achieving high throughput. This paper focuses on the capacity and throughput optimization of 1-hop wireless broadcast with hard deadline constraints. The results could lead to novel protocols with guaranteed QoS for delay-sensitive services for downlink users.

One way of wireless downlink broadcast is to transmit each packet without any coding while bookkeeping the reception status for each individual receiver, respectively. Those packets that are not received successfully for some receivers are retransmitted at a later time. Among these *uncoded schemes*, [1] provides a data broadcast scheduling policy that reduces the average response time for the users, where the response time is the interval between the time when users request an information packet and the time it is broadcast. [2] proposes a scheduling algorithm to broadcast to "impatient" users, where the objective is to both maximize the percentage of requests served and to minimize the mean waiting time. A parameterized algorithm is proposed in [3], which allows the network designer to focus on either the average or the worst waiting time among different users.

Recently, a new class of network-coding-based broadcast schemes emerges, which performs information mixing among different packets before transmission. When there are no deadline constraints, it is well known that broadcasting coded packets can achieve a higher throughput than uncoded policies [4]. Moreover, one can achieve the optimal broadcast capacity (when there are no deadline constraints) using only linear network codes [5]. [6] provides an efficient algebraic framework to characterize the performance of linear network codes, upon which a capacity-achieving random linear network coding scheme is built [7]. Besides the information-theoretic research, practical generation-based network coding schemes have been devised in [8], which further takes into account the lossy channels in a wireless network. [9] considers a 1-hop broadcast channel with erasures and maintains a *source queue* at the base station such that all packets within the source queue are coded together and broadcast to the receivers. With a new coding scheme that carefuly decides which packet to remain in the queue, the necessary buffer size (for the source queue) can be substantially reduced in an effective way. Both the generation-based [8] and the queue-based schemes [9] do not optimize the delay characteristics of the problem. Nonetheless, a commonly referred rule of thumb is that the larger the throughput of the system, the more packets need to be coded together (thus larger generations or longer queues), which in turn causes longer delay as the users need to accumulate a larger number of coded packets before being able to decode a single information packets. Similar to the setting of the 1-hop broadcast session, COPE [10] encodes packets from *different unicast sessions* that enables *1-hop intersession decoding*. The corresponding scheduling and coding policy that decides how to mix packets from different sessions is studied in [11].

In this paper, we are interested in using network coding to improve the throughput for delay-sensitive applications. For such delay-sensitive applications, several network coding works have been proposed based on different delay metrics. For example, the authors of [12]–[14] focus on minimizing delay between the receipt of the coded packets and actual decoding, which is generally referred to as the *decoding delay*. In particular, [12] discusses how different methods of encoding can affect the decoding delay in an error-free network from the information-theoretic perspective. The quantification of gains in delay performance resulting from network coding are studied in [13], and they focus on the completion time of the whole data as the delay metric. [15] focuses on the total transmission delay, i.e., the time interval between the packet arrival at the source and the packet being decoded at the receiver. Their coding module achieves optimal throughput and is conjectured to simultaneously achieve asymptotically minimal delay. [16] proposes a coding scheme assisted by uncoded transmission that reduces the transmission delay.

In constrast to these prior works that focus on minimizing various notions of delay, in this work we focus on the case when each packet has a hard deadline constraint. Within this framework, we characterize the optimal capacity regions (the amount of packets that are received before deadlines) for both uncoded and coded solutions. Our study accounts for the overall transmission delay (i.e., queueing plus coding delays). The setting of this work is mostly related to the following works. Based on the Whittle relaxation for restless bandit problems, [17] considers a similar hard-deadline-constrained, throughput optimization problem for *multiple unicast flows without network coding* and focuses on scheduling/balancing between multiple flows. In comparison, in this work we consider a single broadcast session and we study the capacity for both *coding* and *non-coding policies.* This hard deadline constrained problem with network coding was briefly considered in [14] with an order analysis of the queue-length growth rate. The proposed network coding protocol in our work can be viewed as a generalization of the schemes in [14], [18] for the deadline constrained setting.

The detailed contributions of this work are listed as follows.

- *Uncoded transmission:* A new Markov-decision framework is formulated to analyze the performance of uncoded transmission schemes with hard packet delivery deadlines. A throughput optimal non-coding-based scheduling policy is devised based on finite-horizon dynamic programming (DP), for which the optimal expected throughput can be computed numerically. Since a closed-form expression of the achievable rate of the optimal DP policy is difficult to compute for any finite N, where N is the size of the file of interest, we derive the asymptotic rate of the optimal DP policy as $N \to \infty$. Our results show that although the achievable rate of the optimal DP scheme matches the broadcast channel capacity for the single-user case, it is strictly lower than the optimal capacity in the two-user case. That is, the best non-coding scheme is throughput suboptimal in the deadline-constrained network.
- *Network-coded transmission:* We propose a *universal* network coding scheme under the hard packet deadline constraints, which does not require the knowledge about the delivery rate p of the wireless broadcast channel. (In contrast, the optimal DP-based uncoded scheme requires the knowledge about p before making the decision.) We then prove that the proposed network coding scheme achieves the maximum broadcast channel capacity asymptotically for large N in the 2-user case. Unlike the existing generation-based and queue-based network coding scheme, our results show that a properly designed network coding scheme can achieve the capacity even subject to deadline constraints. We also prove that when N tends to infinity, the total transmission delay (from packet arrival at the base station to decoding at the individual user) remains finite with probability one. Compared to generation-based schemes that require large generation size (and thus large delay) to approach the maximum throughput (in order to average out the randomness of the channel), our scheme attains the maximum channel capacity with finite delay.
- Extensive simulations are conducted, which shows that our predicted asymptotic throughput is closely related to the throughput performance even for small finite N. The simulation results also show that the proposed network-coding scheme not only outperforms the uncoded schemes asymptotically but also for finite N (packets/file) that can be as small as 100.

The rest of the paper is organized as follows: Section II describes the system model and defines the corresponding throughput optimization problem with hard deadline constraints. In Section III, the optimal non-coding scheduling policy is devised based on finite horizon dynamic programming. In addition to the numerical formula of the optimal expected throughput, the closed form expression of the throughput is developed in the asymptotic regime as $N \to \infty$. A novel network coding scheme is proposed in Section IV, and its optimality is proven with a queue-length-based Lyapunov function. In Section V we use simulation to verify the prediction of our theoretic analysis for the cases in which N is small. Some operational insights of the proposed schemes are also discussed in Section V. This work is concluded in Section VI.

II. System Model

We consider the downlink of a single cell in which the base station (BS) broadcasts a file of N packets to multiple users. We assume that time is slotted. Each packet $n = 1, 2, \ldots, N$ has a deadline d_n, after which the packet is no longer useful. To model video streaming applications, we assume that the deadlines of the n-th packet are the same for all users, and are of the form

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

In other words, packet n expires at time slot λn.

We consider random and unreliable wireless channels. That is, a packet broadcast from the BS may be received by all users, a subset of users, or no users at all, depending on the random channel conditions. Suppose a packet is transmitted at the t-th time slot. We use $C_j(t) = 1$ to denote that user j can receive the packet successfully, and $C_j(t) = 0$, otherwise. We assume that $\mathsf{P}(C_j(t) = 1) = p$, and $C_j(t)$ is independently and identically distributed (i.i.d.) across users and time slots. We assume that at the end of each time slot, the BS has perfect feedback whether the packet has been successfully received by each user. Note that based on the feedback, the BS can adapt different transmission strategies and decide whether to send uncoded or coded packet at the next time slot.

If coding is prohibited, the source can only transmit uncoded packets. After receiving the feedback from users in the end of time slot t, the BS may decide to send a new packet or retransmit the old packet at time $t+1$. For multiple users the BS may decide to retransmit even though part of the users may have already received it.

If coding across different packets is allowed, then in one slot, any unexpired packets can be coded together and the BS can choose to broadcast the coded packet to all users. For example, suppose that packet n_1, n_2 and n_3 ($n_1 < n_2 < n_3$) are coded together by random linear network coding and broadcast by the BS. If user 1 has already received two other coded packets generated from packet n_1, n_2 and n_3, then after receiving this new coded packet successfully, it could decode all 3 packets. We assume that the coding coefficients are chosen uniformly and randomly from a sufficiently large Galois field such that if a user receives any 3 coded packets out of the original packets n_1, n_2 and n_3, it could decode the original packets. As will be shown in Section IV, this random coding policy can be further relaxed. We will show that a deterministic linear-coding-based policy that uses only binary XOR operations is sufficient to achieve the capacity of this deadline-constrained system.

Our goal is to design a coding/scheduling policy that maximizes the number of successful (not-expired) packet transmissions. For ease of exposition, in the rest of the paper, we will mainly focus on the case with 2 users. Let $D_j(n) = 1$ if user j (j=1,2) can successfully decode/recover the n-th information packet from all the coded/uncoded packets received before its deadline λn; and $D_j(n) = 0$, otherwise. We define the total number of successes N_{success} as $\sum_{n=1}^{N} D_1(n) + D_2(n)$. Specifically, we would like to maximize the normalized expected throughput given by $\frac{\mathbb{E}\{N_{\text{success}}\}}{2N}$. We are interested in the optimal policies that maximize the normalized expected throughput for both uncoded and coded cases, and study the throughput improvement of network coding subject to hard deadline constraints.

III. The Optimal Policy for Uncoded Transmission

A. A Dynamic-Programming-Based Policy

For the uncoded case, the optimal transmission policy can be solved by formulating the problem as a Markov decision problem. Without loss of generality, we assume that the BS transmits packets sequentially in the order of their deadlines. At each time slot, the BS may choose to retransmit the same packet or choose to drop the packet even before the packet expires. If the current packet expires, the BS always moves to the next packet.

At the end of each time slot, denote the state of the Markov Decision problem as $\vec{x}(t) = (T, n, x_1, x_2)$, where $T \overset{\Delta}{=} \lambda N - t$ is the number of remaining time slots before λN (the time when the last packet expires). The variable n ($1 \leq n \leq N$) denotes the index of the first packet that the BS may transmit in future time slots. In other words, the BS will not transmit any packets with index $< n$ in future time slots from $t+1$ to λN. x_1 and x_2 are two Bernoulli variables. $x_1 = 1$ denotes that packet n has already been received by user 1, $x_1 = 0$, otherwise. $x_2 = 1$ denotes that packet n has already been received by user 2. $x_2 = 0$, otherwise. Since it is useless to transmit a packet that has expired already, we also require that $\lambda n > t$. Therefore, the set of all valid states become

$$\vec{X} = \\ \{(T, n, x_1, x_2) | n\lambda > N\lambda - T, 1 \leq n \leq N, x_1, x_2 \in \{0, 1\}\}.$$

The terminal state of the Markov decision problem are of the forms $(\cdot, N+1, 0, 0)$ (when there are no packets remained to be sent at the BS), or $(0, N+1, \cdot, \cdot)$ (when all packets have expired). The starting state is $(\lambda N, 1, 0, 0)$.

Let $u(\vec{x})$ denote the action taken by the BS at the state $\vec{x}$. $u(\vec{x}) = 1$ means that the BS sends the packet n in the next time slot (i.e., the $(\lambda N - T + 1)$-th time slot). $u(\vec{x}) = 0$ means that the BS decides that packet n is not interesting anymore and skip to the next packet. Once the BS decides to skip to the next packet, packet n will no longer be transmitted in any future time slot. For future reference, we say that when $u(\vec{x}) = 0$ the *BS drops the packet* n.

Define $C_j(\vec{x}) = C_j(\lambda N - T + 1)$ as the channel realization in the coming time slot. Recall that $C_j(\vec{x}) = 1$ indicates that the packet sent over the $(\lambda N - T + 1)$-th time slot will be successfully received, and $C_j(\vec{x}) = 0$ indicates that the packet is erased during transmission. By our erasure channel setting, $\mathsf{P}(C_j(\vec{x}) = 1) = p$ and $C_1(\vec{x})$ is independent of $C_2(\vec{x})$. We also define the joint vector $\vec{C}(\vec{x}) = (C_1(\vec{x}), C_2(\vec{x}))$. When it is clear from the context, we often drop the input argument $\vec{x}$ and use u and $\vec{C}$ as shorthand notations to denote the action and the channel realization, respectively, for a given state $\vec{x}$.

Let $\vec{x'}$ be the next state of the Markov decision problem. Then $\vec{x'}$ can be written as a function of $\vec{x}$, u and $\vec{C}$

$$\vec{x'} = (T', n', x'_1, x'_2) = f(\vec{x}, u, \vec{C}). \tag{1}$$

More specifically, when $u = 0$ (the BS decides to drop the packet n), we have $\vec{x'} = (T, n+1, 0, 0)$ since no physical transmission is made and we still have T more time slots to transmit packets $n+1$ to N. When $u = 1$ (the BS decides to transmit the packet n) the next stage depends on the channel realization $\vec{C}$. If both $(x_1 \vee C_1) = 1$ and $(x_2 \vee C_2) = 1$, where $\vee$ is the binary OR operation, then the packet n is received successfully by both users. Therefore, the BS will move on to the next packet and $\vec{x'} = (T-1, n+1, 0, 0)$. If one of them is not satisfied, then we have two different sub-cases: If $N\lambda - (T-1) < n\lambda$, then packet n has not expired after the transmission. Therefore, we can retransmit packet n. The next state becomes $\vec{x'} = (T-1, n, x_1 \vee C_1, x_2 \vee C_2)$. If $N\lambda - (T-1) = n\lambda$, then the packet n has expired after this transmission. Even though one of the users has not received this packet successfully, we still have to move on to the next packet. We thus have $\vec{x'} = (T-1, n+1, 0, 0)$. The above relationship of $\vec{x'}$ and the current state/decision can be summarized by a single $f(\cdot,\cdot,\cdot)$ function as in (1).

For action u, let $L(\vec{x}, u, \vec{C})$ denote the reward (the number of successfully received packets) with current state $\vec{x}$ and the channel realization $\vec{C}$. When $u = 1$, the reward depends on the number of users that receives an *additional* packet (i.e., $x_j = 0$ and $C_j = 1$). We thus have

$$\begin{aligned} &L(\vec{x}, 1, \vec{C}) \\ &= \begin{cases} 2 & \text{if both users receive additional packets} \\ 1 & \text{if only one user receives an additional packet} \\ 0 & \text{otherwise} \end{cases}. \end{aligned}$$

Use $J(\vec{x})$ to denote the maximum expected cumulated reward when starting from state $\vec{x}$. For terminal states $\vec{x}$ we have $J(\vec{x}) = 0$. For non-terminal states $\vec{x}$, $J(\vec{x})$ can be computed iteratively from the terminal states back to the starting state by the following Bellman's equation:

$$J(\vec{x}) = \max_{u=0,1} \mathbb{E}_{\vec{C}}\{L(\vec{x}, u, \vec{C}) + J(f(\vec{x}, u, \vec{C}))\}. \tag{2}$$

where

$$\mathbb{E}_{\vec{C}}\{L(\vec{x}, u, \vec{C})\} = \begin{cases} 2p & \text{if } u = 1,\ x_1 = x_2 = 0 \\ p & \text{if } u = 1,\ x_1 \neq x_2 \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

In addition to computing the cumulative expected reward, (2) can also be used by the BS to compute the optimal decision $u(\vec{x})$. Namely, at any time t, depending on the corresponding state $\vec{x}$, the BS makes its decision by choosing the maximizing u of (2). The optimal decision $u(\vec{x})$ can also be expressed in a threshold form. For any state $\vec{x}$, define a function $g(\vec{x})$ as follows. If $x_1 = x_2 = 0$, then

$$\begin{aligned} g(\vec{x}) \overset{\Delta}{=}\ & J(T, n+1, 0, 0) - (1-p)^2 J(T-1, n, 0, 0) \\ & - 2p(1-p) J(T-1, n, 1, 0) - p^2 J(T-1, n+1, 0, 0). \end{aligned}$$

If $x_1 \neq x_2$, then

$$\begin{aligned} g(\vec{x}) \overset{\Delta}{=}\ & J(T, n+1, 0, 0) - (1-p) J(T-1, n, 1, 0) \\ & - pJ(T-1, n+1, 0, 0). \end{aligned}$$

Note that this $g(\vec{x})$ function can be pre-computed before transmission. In the beginning of each time slot $(t+1)$, the BS makes the decision based on

$$u(\vec{x}) = \begin{cases} 1 & \text{if } \mathbb{E}_{\vec{C}}\{L(\vec{x}, 1, \vec{C})\} \geq g(\vec{x}) \\ 0 & \text{otherwise} \end{cases},$$

where $\mathbb{E}_{\vec{C}}\{L(\vec{x}, 1, \vec{C})\}$ is computed from (3). The proof of this threshold-based decision rule is straightforward from (2) and by noting that $J(T, n, 1, 0) = J(T, n, 0, 1)$ for any T and n due to the symmetry of the channels for both users.

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. 1. Illustration of w_i. "E" means erasure, and "rec." means successfully received by a user

We refer to the optimal decisions $u(\vec{x})$ obtained from the above iterative equations as the DP policy, which is throughput optimal for uncoded transmission. Although the DP policy can be efficiently computed numerically, a closed form expression of the expected reward is hard to obtain for finite N. In the following subsections, we focus on the normalized throughput of the DP policy for asymptotically large N.

B. 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 deadline λN (instead of individual deadlines λn). The maximum throughput under this relaxed setting thus serves as an upper bound for the original problem. To that end, we first categorize the packet broadcast at a given time t into three types: types-0, 1, and 2, which indicates how many users have received this packet in the previous time slots ($[1, t-1]$). Consider any transmission policy. Let w_0, w_1 and w_2 denote the numbers of time slots that are used to transmit packets that have been received by 0, 1 and 2 users, respectively.

Fig. 1 illustrates the construction of w_0 to w_2 for a given policy and channel realization. Before all the transmissions, all w_i are set to be 0. In Fig. 1, 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 increments. 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 $\overline{w}_0$, $\overline{w}_1$ and $\overline{w}_2$ denote the expectation of w_0, w_1 and w_2, respectively. When the BS transmits a new packet to both users, the expected reward in the time slot is exactly $2p$. And when the BS transmits a packet that has been received by one user, the expected reward in the time slot is p. Transmitting a packet that has already been received by 2 users would not improve the throughput. Using the optional sampling theorem for Martingales, we can show that the expected total number of successes is $2p\overline{w}_0 + p\overline{w}_1$.

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

$$\overline{w}_1 p \leq \overline{w}_0 2p(1-p). \tag{4}$$

By noting that $w_0 + w_1 + w_2 = \lambda N$ and by the conservation law of types-0 and 2 packets in a similar way as in (4), we have

$$\begin{aligned}
\mathbb{E}\{N_{\text{success}}\} =& 2p\overline{w}_0 + p\overline{w}_1 + 0 \cdot \overline{w}_2 && (5)\\
\text{s.t.}\quad & \overline{w}_0 + \overline{w}_1 + \overline{w}_2 \leq \lambda N \\
& \overline{w}_0(2p-p^2) \leq N \\
& \overline{w}_1 p \leq \overline{w}_0(2p(1-p)) \\
& \overline{w}_0, \overline{w}_1, \overline{w}_2 \geq 0 && (6)
\end{aligned}$$

Let μ_i be the Lagrangian multiplier of the above problem. The corresponding KKT conditions become

$$\begin{aligned}
& -2p + \mu_1 + \mu_2(2p-p^2) - \mu_3 2p(1-p) - \mu_4 = 0 \\
& -p + \mu_1 + \mu_3 p - \mu_5 = 0 \\
& \mu_1 - \mu_6 = 0 \\
& \mu_1(\overline{w}_0 + \overline{w}_1 - \lambda N) + \mu_2(\overline{w}_0(2p-p^2) - N) && (7)\\
& \quad + \mu_3(-\overline{w}_0(2p(1-p)) + \overline{w}_1 p) \\
& \quad - \mu_4 w_0 - \mu_5 \overline{w}_1 - \mu_6 \overline{w}_2 = 0 \\
& -\mu_4 \overline{w}_0 \leq 0, \ \ -\mu_5 \overline{w}_1 \leq 0, \ \ -\mu_6 \overline{w}_2 \leq 0 \\
& \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6 \geq 0
\end{aligned}$$

We are now ready to derive the closed form solution of the optimization problem in (5), depending on differenet values of p.

Case 1: $\frac{\lambda+1-\sqrt{\lambda^2+1-\lambda}}{\lambda} < p \leq 1$. We choose $\mu_1 = 0, \mu_2 = 2, \mu_3 = 1, \mu_4 = 0, \mu_5 = 0, \mu_6 = 0$, $\overline{w}_0 = \frac{N}{2p-p^2}$, $\overline{w}_1 = \frac{2(1-p)N}{2p-p^2}$, and $\overline{w}_2 = 0$. It is easy to check that these choices of μ_i and w_i satisfy the KKT conditions in (7). Therefore, the maximum value of (5) is

$$2p\left(\frac{N}{2p-p^2}\right) + p\left(\frac{2(1-p)N}{2p-p^2}\right) = 2N.$$

Case 2: $\frac{\lambda-\sqrt{\lambda^2-\lambda}}{\lambda} < p \leq \frac{\lambda+1-\sqrt{\lambda^2+1-\lambda}}{\lambda}$, we choose $\mu_1 = p, \mu_2 = \frac{1}{2-p}, \mu_3 = 0, \mu_4 = 0, \mu_5 = 0, \mu_6 = p$, $\overline{w}_0 = \frac{N}{2p-p^2}$, $\overline{w}_1 = \lambda N - \frac{N}{2p-p^2}$, and $\overline{w}_2 = 0$. It can be checked that these μ_i and w_i values satisfy the KKT conditions in (7). Therefore, the maximum value of (5) is

$$2p\left(\frac{N}{2p-p^2}\right) + p\left(\lambda N - \frac{N}{2p-p^2}\right) = \frac{2N}{2}\left(\lambda p + \frac{1}{2-p}\right).$$

Case 3: $0 < p < \frac{\lambda-\sqrt{\lambda^2-\lambda}}{\lambda}$, we choose $\mu_1 = 2p, \mu_2 = 0, \mu_3 = 0, \mu_4 = 0, \mu_5 = p, \mu_6 = 2p$, $\overline{w}_0 = \lambda N$, $\overline{w}_1 = 0$, and $\overline{w}_2 = 0$. these μ_i and w_i values satisfy the KKT conditions in (7). Therefore, the maximum value of (5) is

$$2p\left(\lambda N\right) + p\left(0\right) = 2\lambda p N.$$

Since the above constraints hold for any policy, we can thus upper bound the best achievable rate for uncoded transmission by maximizing (5) subject to the constraints in (6). A closed-form solution to this linear program then produces the following upper bound

$$\begin{aligned}
& \frac{\mathbb{E}\{N_{\text{success}}\}}{2N} \\
& \leq \begin{cases} 1 & \text{if } \frac{\lambda+1-\sqrt{\lambda^2+1-\lambda}}{\lambda} < p \leq 1 \\ \frac{1}{2}(\lambda p + \frac{1}{2-p}) & \text{if } \frac{\lambda-\sqrt{\lambda^2-\lambda}}{\lambda} < p \leq \frac{\lambda+1-\sqrt{\lambda^2+1-\lambda}}{\lambda} \\ \lambda p & \text{if } 0 < p \leq \frac{\lambda-\sqrt{\lambda^2-\lambda}}{\lambda} \end{cases}.
\end{aligned} \tag{8}$$

C. An Inner Bound for the Optimal Throughput When $N \to \infty$

Next we will show that the upper bound in (8) can be achieved when $N \to \infty$. 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 and X_n^2. Let X_n^1 denote the number of additional time slots it takes before the BS-to-user-1 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 $\mathsf{P}\left(X_n^1 = k\right) = (1-p)^{k-1}p$ for all n. We define X_n^2 by symmetry. Let $Y_n = \min(X_n^1, X_n^2)$ denote the number of time slots it takes before at least one user has received at least one packet. Let $Z_n = \max(X_n^1, X_n^2)$ denote the number of time slots before both users receive at least one packet. By simple probability computation, we have $\mathbb{E}\{Y_n\} = \frac{1}{2p-p^2}$ and $\mathbb{E}\{Z_n\} = \frac{1}{p^2-2p} + \frac{2}{p}$.

Consider the following transmission policies. In the first policy, for any ongoing packet (say packet n), repeatedly transmit it until both users receive it or the packet expires. Then move to packet $n+1$, repeatedly transmit packet $n+1$, until both users receive it or the packet expires, and so on.

We denote this policy by $\pi_{\max}$. The second policy $\pi_{\min}$ is similar to policy $\pi_{\max}$, and the difference is that the BS keeps transmitting the ongoing packet until the packet is received by at least one user. Then the BS moves on to the next packet.

We now consider several schemes that perform *random mixtures of* $\pi_{\min}$, $\pi_{\max}$, 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 three options: transmit a packet based on policy $\pi_{\max}$, or based on policy $\pi_{\min}$, or the BS can simply drop the current packet and move to the next. The BS chooses *randomly and independently* among these three 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 three sub-policies. Let $\Gamma_{\min,n}$ and $\Gamma_{\max,n}$ denote the events that the BS chooses to transmit packet n by policies $\pi_{\min}$ and $\pi_{\max}$, respectively. Similar to Section III-B, we need to discuss three different cases depending on the value of p.

Case 1: $\frac{\lambda+1-\sqrt{\lambda^2+1}-\lambda}{\lambda} < p \leq 1$. In this case, we set $\mathsf{P}(\Gamma_{\max,n}) = 1$ and $\mathsf{P}(\Gamma_{\min,n}) = 0$. In other words, $\pi_{\max}$ is used for all packets in this case.

Case 2: $\frac{\lambda-\sqrt{\lambda^2-\lambda}}{\lambda} < p \leq \frac{\lambda+1-\sqrt{\lambda^2+1}-\lambda}{\lambda}$. In this case, the random mixture of policies is given by

$$\begin{aligned}
&\mathsf{P}\left(\Gamma_{\min,n}\right) = \frac{-p^2\lambda + 2p(\lambda+1) - 3}{2p-2} - \frac{\epsilon_1}{2}\\
&\mathsf{P}\left(\Gamma_{\max,n}\right) = \frac{p^2\lambda - 2p\lambda + 1}{2p-2} - \frac{\epsilon_1}{2}\\
&\text{and } \mathsf{P}\left(\text{BS decides to drop packet } n\right) = \epsilon_1,\\
&\quad\text{where } \epsilon_1 > 0 \text{ is a small constant.} \qquad (9)
\end{aligned}$$

Case 3: $0 < p \leq \frac{\lambda-\sqrt{\lambda^2-\lambda}}{\lambda}$. In this case, we set $\mathsf{P}(\Gamma_{\max,n}) = 0$ and $\mathsf{P}(\Gamma_{\min,n}) = \lambda(2p-p^2) - \epsilon_2$, and the packet dropping probability being $1 - \lambda(2p-p^2) + \epsilon_2$ for some $\epsilon_2 > 0$.

In the following, we provide a detailed proof for the most complicated case: Case 2 in Section III-B. Case 1 and 3 can be viewed as simple extension of Case 2. We will show that the above random mixture policy achieves the upper bound (8) when N is sufficiently large.

Let W_n be the number of transmissions of the n-th packet. W_n can be defined iteratively as follows:

$$\begin{aligned}
&W_1 = \min\left(Y_1\mathbf{1}_{\Gamma_{\min,1}} + Z_1\mathbf{1}_{\Gamma_{\max,1}}, \lambda\right),\\
&\forall n = 2, \ldots N,\\
&W_n = \min\left(Y_n\mathbf{1}_{\Gamma_{\min,n}} + Z_n\mathbf{1}_{\Gamma_{\max,n}}, \lambda n - \sum_{k=1}^{n-1} W_k\right).
\end{aligned}$$

where $\mathbf{1}_{\{\cdot\}}$ is the indicator function. Since the n-th packet may expire even before one or both users receive the desired packet (depending on whether it is $\pi_{\min}$ or $\pi_{\max}$ policy being chosen), the number of times that the n-th packet is sent is the minimum of the two. We thus have the above iterative equations for W_n.

Define $S_n \triangleq \sum_{i=1}^{n-1} W_i + Y_n\mathbf{1}_{\Gamma_{\min,n}} + Z_n\mathbf{1}_{\Gamma_{\max,n}}$ and $W_1^{n-1} \triangleq \{W_1, W_2, \ldots, W_{n-1}\}$. Since $Y_n\mathbf{1}_{\Gamma_{\min,n}} + Z_n\mathbf{1}_{\Gamma_{\max,n}}$ is independent of W_1^{n-1}, we have

$$\begin{aligned}
&\mathbb{E}\left\{e^{\tau S_n}\right\} = \mathbb{E}\left\{\mathbb{E}\left\{e^{\tau S_n}|W_1^{n-1}\right\}\right\}\\
&= \mathbb{E}\left\{e^{(\sum_{i=1}^{n-1} W_i)\tau}\mathbb{E}\left\{e^{\tau(Y_n\mathbf{1}_{\Gamma_{\min,n}} + Z_n\mathbf{1}_{\Gamma_{\max,n}})}|W_1^{n-1}\right\}\right\}\\
&= \mathbb{E}\left\{e^{\sum_{i=1}^{n-1} W_i\tau}\right\}\mathbb{E}\left\{e^{\tau(Y_n\mathbf{1}_{\Gamma_{\min,n}} + Z_n\mathbf{1}_{\Gamma_{\max,n}})}\right\}. \qquad (10)
\end{aligned}$$

Since by definition $W_{n-1} \leq Y_{n-1}\mathbf{1}_{\Gamma_{\min,n-1}} + Z_{n-1}\mathbf{1}_{\Gamma_{\max,n-1}}$, we have

$$\begin{aligned}
&\sum_{i=1}^{n-1} W_i\\
&\quad \leq S_{n-1} \triangleq \sum_{i=1}^{n-2} W_i + Y_{n-1}\mathbf{1}_{\Gamma_{\min,n-1}} + Z_{n-1}\mathbf{1}_{\Gamma_{\max,n-1}}.
\end{aligned} \qquad (11)$$

By iteratively applying (10), we have

$$\begin{aligned}
\mathbb{E}\{e^{\tau S_n}\} &\leq \prod_{i=1}^{n} \mathbb{E}\left\{e^{\tau(Y_i\mathbf{1}_{\Gamma_{\min,i}} + Z_i\mathbf{1}_{\Gamma_{\max,i}})}\right\}\\
&= \left(\mathbb{E}\left\{e^{\tau(Y_n\mathbf{1}_{\Gamma_{\min,n}} + Z_n\mathbf{1}_{\Gamma_{\max,n}})}\right\}\right)^n,
\end{aligned}$$

where the last equality follows from that $Y_i\mathbf{1}_{\Gamma_{\min,i}} + Z_i\mathbf{1}_{\Gamma_{\max,i}}$ has the same marginal distribtuion for all i. Since $\mathbb{E}\left\{Y_n\mathbf{1}_{\Gamma_{\min,n}} + Z_n\mathbf{1}_{\Gamma_{\max,n}}\right\} = \lambda - \frac{\epsilon_1}{p}$, we can choose $\gamma = \lambda - \epsilon_1 > \mathbb{E}\left\{Y_n\mathbf{1}_{\Gamma_{\min,n}} + Z_n\mathbf{1}_{\Gamma_{\max,n}}\right\}$. It can be easily checked that with our construction of Y_n and Z_n, there exists a $\delta > 0$ such that $\mathbb{E}\left\{e^{\tau\left(Y_n\mathbf{1}_{\Gamma_{\min,n}} + Z_n\mathbf{1}_{\Gamma_{\max,n}}\right)}\right\}$ exists for all τ in the open interval $(-\delta, \delta)$ containing 0. Combining with our choice of $\gamma > \mathbb{E}\left\{Y_n\mathbf{1}_{\Gamma_{\min,n}} + Z_n\mathbf{1}_{\Gamma_{\max,n}}\right\}$, there must exist a small $\widetilde{\tau}_1 > 0$, such that

$$\mathbb{E}\left\{e^{\widetilde{\tau}_1\left(Y_n\mathbf{1}_{\Gamma_{\min,n}} + Z_n\mathbf{1}_{\Gamma_{\max,n}}\right)}\right\} < e^{\widetilde{\tau}_1\gamma}.$$

Conditioning on that the BS chooses policy $\pi_{\min}$, the probability that packet n can be received by at least one user is

$$\begin{aligned}
&\mathsf{P}\left(\text{at least one user receives pkt } n|\Gamma_{\min,n}\right)\\
&= \mathsf{P}\left(W_n = Y_n|\Gamma_{\min,n}\right)\\
&= 1 - \mathsf{P}\left(S_n > \lambda n|\Gamma_{\min,n}\right) \geq 1 - \frac{1}{\mathsf{P}\left(\Gamma_{\min,n}\right)}\mathsf{P}\left(S_n \geq \lambda n\right)\\
&\geq 1 - \frac{1}{\mathsf{P}\left(\Gamma_{\min,n}\right)}\min_{\tau>0}\frac{\mathbb{E}\{e^{\tau S_n}\}}{e^{\lambda n\tau}}\\
&\geq 1 - \frac{1}{\mathsf{P}\left(\Gamma_{\min,n}\right)}\min_{\tau>0}\frac{\mathbb{E}\left\{e^{\tau(Y_n\mathbf{1}_{\Gamma_{\min,n}} + Z_n\mathbf{1}_{\Gamma_{\max,n}})}\right\}^n}{e^{\lambda n\tau}}\\
&\geq 1 - \frac{e^{\gamma n\widetilde{\tau}_1}}{\mathsf{P}\left(\Gamma_{\min,n}\right)e^{\lambda n\widetilde{\tau}_1}} \geq 1 - \frac{e^{-\widetilde{\tau}_1\epsilon_1 n}}{\mathsf{P}\left(\Gamma_{\min,n}\right)}.
\end{aligned}$$

Similarly, conditioning on that the BS chooses policy $\pi_{\max}$, the probability that packet n can be received by both users is lower bounded by

$$\mathsf{P}\left(\text{both users receive packet } n|\Gamma_{\max,n}\right)$$

$$= \mathsf{P}\left(W_n = Z_n | \Gamma_{\max,n}\right) \geq 1 - \frac{e^{-\widetilde{\tau}_1 \epsilon_1 n}}{\mathsf{P}\left(\Gamma_{\max,n}\right)}.$$

Let T_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.[1] We thus have

$$\begin{aligned}
&\mathsf{P}\left(T_n = 2 | \Gamma_{\min,n}\right) = \sum_{k=1}^{\infty} \mathsf{P}\left(X_n^1 = X_n^2 = k\right) = \frac{p^2}{2p - p^2}, \\
&\mathsf{P}\left(T_n = 1 | \Gamma_{\min,n}\right) = 1 - \frac{p^2}{2p - p^2}, \\
&\mathsf{P}\left(T_n = 2 | \Gamma_{\max,n}\right) = 1.
\end{aligned}$$

Let A_n denote the event that when packet n is transmitted, only one user has received it (unexpired); and let B_n denote the event that when packet n is transmitted, both users have received it (unexpired). Then by the union bound, we have

$$\begin{aligned}
\mathsf{P}\left(A_n | \Gamma_{\min,n}\right) =& \mathsf{P}\left(T_n = 1, W_n = Y_n | \Gamma_{\min,n}\right) \\
\geq& 1 - \frac{p^2}{2p - p^2} - \frac{e^{-\widetilde{\tau}_1 \epsilon_1 n}}{\mathsf{P}\left(\Gamma_{\min,n}\right)}, \\
\mathsf{P}\left(B_n | \Gamma_{\min,n}\right) =& \mathsf{P}\left(T_n = 2, W_n = Y_n | \Gamma_{\min,n}\right) \\
\geq& \frac{p^2}{2p - p^2} - \frac{e^{-\widetilde{\tau}_1 \epsilon_1 n}}{\mathsf{P}\left(\Gamma_{\min,n}\right)}, \\
\mathsf{P}\left(B_n | \Gamma_{\max,n}\right) =& \mathsf{P}\left(T_n = 2, W_n = Z_n | \Gamma_{\max,n}\right) \\
\geq& 1 - \frac{e^{-\widetilde{\tau}_1 \epsilon_1 n}}{\mathsf{P}\left(\Gamma_{\max,n}\right)}.
\end{aligned}$$

The expected throughput for packet n thus satisfies,

$$\begin{aligned}
&\mathbb{E}\{D_1(n) + D_2(n)\} \\
&= \mathsf{P}\left(A_n\right) + 2\mathsf{P}\left(B_n\right) \geq \mathsf{P}\left(A_n | \Gamma_{\min,n}\right) \mathsf{P}\left(\Gamma_{\min,n}\right) \\
&\quad + 2\mathsf{P}\left(B_n | \Gamma_{\min,n}\right) \mathsf{P}\left(\Gamma_{\min,n}\right) + 2\mathsf{P}\left(B_n | \Gamma_{\max,n}\right) \mathsf{P}\left(\Gamma_{\max,n}\right) \\
&\geq \frac{\lambda p^3 - 3\lambda p^2 + (2\lambda - 1)p + 1}{(p-2)(p-1)} - \frac{3p - p^2}{2p - p^2}\epsilon_1 - 5e^{-\widetilde{\tau}_1 \epsilon_1 n}.
\end{aligned}$$

Since $\sum_{n=1}^{\infty} e^{-\widetilde{\tau}_1 \epsilon_1 n} < \infty$, we have

$$\begin{aligned}
&\lim_{N \to \infty} \frac{\mathbb{E}\{N_{\text{success}}\}}{2N} = \lim_{N \to \infty} \frac{\sum_{n=1}^{N} \mathbb{E}\{D_1(n) + D_2(n)\}}{2N} \\
&= \frac{-\lambda p^3 + 3\lambda p^2 - (2\lambda - 1)p - 1}{(2p-2)(2-p)} - \frac{\epsilon_1}{2} \frac{3p - p^2}{2p - p^2} \\
&= \frac{1}{2}(\lambda p + \frac{1}{2-p}) - \frac{\epsilon_1}{2} \frac{3p - p^2}{2p - p^2}. \qquad (12)
\end{aligned}$$

By choosing a sufficiently small ϵ_1, the sub-optimal random mixture scheme achieves the capacity upper bound in (8) when p satisfis the Case 2 condition.

Now we will show that the random mixture policy can also achieve the upper bound when $N \to \infty$ in Case 1, that is when $\frac{\lambda+1-\sqrt{\lambda^2+1-\lambda}}{\lambda} < p \leq 1$. Recall that we have set $\mathsf{P}(\Gamma_{\max,n}) = 1$ and $\mathsf{P}(\Gamma_{\min,n}) = 0$. Since $\mathbb{E}\{Z_i\} < \lambda$, suppose $\mathbb{E}\{Z_i\} = \lambda - \epsilon_2 < \lambda$, ϵ_2 is a positive number. Reassign $\gamma = \lambda - \frac{\epsilon_2}{2}$. Repeat the steps as above proof, we will have $\mathsf{P}(B_n | \Gamma_{\max,n}) \geq 1 - e^{-\frac{\epsilon_2}{2} n \widetilde{\tau}_2}$. (Note that in Case 1 we do not need to consider $\mathsf{P}(A_n | \Gamma_{\min,n})$ and $\mathsf{P}(B_n | \Gamma_{\min,n})$, since $\mathsf{P}(\Gamma_{\min,n}) = 0$.) Then

$$\lim_{N \to \infty} \frac{\mathbb{E}\{N_{\text{success}}\}}{2N} \geq 1$$

For Case 3, that is $0 < p \leq \frac{\lambda - \sqrt{\lambda^2 - \lambda}}{\lambda}$, we have set $\mathsf{P}(\Gamma_{\max,n}) = 0$ and $\mathsf{P}(\Gamma_{\min,n}) = \lambda(2p - p^2) - \epsilon_3$, and the packet dropping probability being $1 - \lambda(2p - p^2) + \epsilon_3$ for some $\epsilon_3 > 0$. We reassign $\gamma = \lambda - \frac{\epsilon_3}{2p}$. Then applying the similar techniques we can have

$$\begin{aligned}
P(A_n | \Gamma_{\min,n}) &\geq 1 - \frac{p^2}{2p - p^2} - \frac{e^{-\frac{\epsilon_3 n \widetilde{\tau}_3}{2p}}}{P(\Gamma_{\min,n})} \\
P(B_n | \Gamma_{\min,n}) &\geq \frac{p^2}{2p - p^2} - \frac{e^{-\frac{\epsilon_3 n \widetilde{\tau}_3}{2p}}}{P(\Gamma_{\min,n})}
\end{aligned}$$

(Note that in Case 3 we do not need to consider $\mathsf{P}(A_n | \Gamma_{\max,n})$ and $\mathsf{P}(B_n | \Gamma_{\max,n})$, since $\mathsf{P}(\Gamma_{\max,n}) = 0$.) Therefore,

$$\begin{aligned}
&\lim_{N \to \infty} \frac{\mathbb{E}\{N_{\text{success}}\}}{2N} = \lim_{N \to \infty} \frac{\sum_{n=1}^{N} \mathbb{E}\{D_1(n) + D_2(n)\}}{2N} \\
&= p\lambda - \frac{\epsilon_3 p}{2p - p^2}
\end{aligned}$$

By choosing a sufficiently small ϵ_3, the sub-optimal random mixture scheme achieves the capacity upper bound in (8) when p satisfies the Case 3 condition.

We have analyzed the asymptotic throughput of the above mixture policies with the individual deadline constraints (λn for the n-th packet). Although the random mixture policy chooses one of the three sub-policies independently for each packet, the deadline constraints impose dependency between the duration of transmitting the n-th packet and the duration of transmitting the $n+1$-th packet, which is the key difficulty of the analysis. Nonetheless, we show that the expected throughput of the above schemes approaches the upper bound in (8) when $N \to \infty$. (8) is thus the closed form expression of the asymptotic capacity for the optimal uncoded DP policy.

IV. A Simple And Asymptotically Optimal Policy for Coded Transmission

In this section, we study the use of network coding for the streaming broadcast problem under hard deadline constraints. We will propose a novel network coding scheme and use a new Lyapunov function to prove that it is asymptotically throughput-optimal when $N \to \infty$. Moreover, the asymptotic capacity of the network coded transmission is shown to be strictly greater than that of the uncoded schemes. Jointly the results in Sections III and IV quantify the throughput improvement of network coding for deadline-constrained systems.

A. A Simple Network Coding Policy

We first propose a novel transmission scheme that uses network coding. (A similar scheme was proposed in [14] and

[1] If $S_n \leq \lambda n$, 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 T_n defined herein does not distinguish these two scenarios only counts the users receiving at least one additional packet within the given interval.

[18]. However, they do not consider hard deadline constraints, and do not carry out the corresponding capacity analysis.)

Define L_j, $j = 1, 2$, to be the list of unexpired packets that user j has not received/decoded but the other user has. For example, suppose we transmit packet 1 in the first time slot and only user 1 receives it. Then $L_2 = \{1\}$ and $L_1 = \emptyset$. If in the second time slot, we transmmit packet 2 and user 2 receives it, then $L_2 = \{1\}$ and $L_1 = \{2\}$, assuming neither packets 1 and 2 have expired. The simple network-coded scheme is described by the following pseudo-code in which the t variable in the FOR-LOOP is the time slot that is currently under consideration and n is an auxiliary variable stored/used by the BS.

```
Set $n \leftarrow 1$, $L_1 \leftarrow \emptyset$ and $L_2 \leftarrow \emptyset$.
for $t = 1$ to $\lambda N$ do
    Remove all the expired entries in $L_1$ and $L_2$ (i.e., those with indices strictly smaller than $\frac{t}{\lambda}$).
    if $n \leq N$ then
        if $L_1 \neq \emptyset$ and $L_2 \neq \emptyset$ then
            In this case, a coding opportunity arises. We send a coded packet by binary XORing two packets $n_1$ and $n_2$, where $n_1$ and $n_2$ are the oldest[2] packets from $L_1$ and from $L_2$, respectively.
        else
            Send an uncoded packet $n$.
            if the uncoded packet is received by at least one user then
                $n \leftarrow n + 1$.
            end if
        end if
    else
        Choose the oldest packet $i$ in $L_1 \cup L_2$, and send packet $i$ uncodedly.
    end if
    UPDATE $L_1$ and $L_2$ by the feedback received from the two users at the end of time $t$.
end for
```

The above pseudo-code for the proposed policy is self-explanatory. The only subroutine "UPDATE L_1 and L_2" is described as follows, depending on whether the BS sends a coded/uncoded packet. Suppose that the BS transmits a coded packet combining packets n_1 and n_2. If user j receives that packet, then user j can *decode* packet n_j. Hence we set $L_j \leftarrow L_j \backslash n_j$. Suppose that the BS transmits an uncoded packet n as in Line IV-A. If only one of the users receives it, say user 1, then $L_2 \leftarrow L_2 \cup \{n\}$ and L_1 remains intact. If only user 2 receives it, then $L_1 \leftarrow L_1 \cup \{n\}$ and L_2 remains unchanged. For any other cases, both L_1 and L_2 remain intact. If the BS transmits an uncoded packet i, say $i \in L_j$, and user j receives it successfully, then we set $L_j \leftarrow L_j \backslash i$.

Remark: The proposed coding scheme does not require the knowledge about the p value, and thus is a universal scheme for any p. For comparison, with uncoded transmission, both the DP policy or the random mixture policy in Sections III-A and III-C require the knowledge about p.

In the following subsection, we quantify the asymptotic throughput of the above scheme and show that it achieves asymptotically the capacity of any coded/uncoded transmission policy.

B. An Upper Bound on the Optimal Throughput with Coded Transmission

We first derive an upper bound for the optimal throughput of any coded/uncoded scheme subject to deadline constraints. Recall that $C_j(t) = 1$ if user j successfully receives a packet transmitted at time slot t. We then note that the total number of packets that all users can recover/decode is upper bounded by the total number of transmitted coded/uncoded packets that they receive successfully. Therefore,

$$\mathbb{E}\{N_{\text{success}}\} \leq \mathbb{E}\left\{\sum_{t=1}^{\lambda N}\sum_{j=1}^{2} C_j(t)\right\} = 2\lambda N p.$$

Further, since the best scenario is that each user can recover/decode all N information packets, we have $\mathbb{E}\{N_{\text{success}}\} \leq 2N$. Jointly we have

$$\frac{\mathbb{E}\{N_{\text{success}}\}}{2N} \leq \min(\lambda p, 1). \tag{13}$$

Note that the upper bound is not only simpler, but also strictly larger than the upper bound of the uncoded throughput in (8). We next show that our proposed scheme can asymptotically achieve the above upper bound.

C. Asymptotic Throughput Optimality of the Proposed Coding Scheme

The capacity in (13) contains two cases: $0 < p < \frac{1}{\lambda}$ and $\frac{1}{\lambda} \leq p \leq 1$. We first focus on the case in which $p < \frac{1}{\lambda}$ and provide the intuition why this scheme can achieve the optimal throughput asymptotically. Recall that there are N packets to transmit. Let t_N be the last time slot before the auxiliary variable n stored at the BS becomes N. We then observe that during the $[1, t_N]$ interval, the BS either transmits an uncoded packet n that is *new* to both users, or transmits a coded packet $n_1 \oplus n_2$ that is *innovative* to both users. Therefore the expected reward for each time slot is exactly $2p$. The total expected reward during the $[1, t_N]$ interval is thus $t_N 2p$. Recall that when $p < \frac{1}{\lambda}$, the capacity upper bound is λp. Hence, in order to approach the upper bound, we would like to show that $t_N \approx \lambda N$. In other words, almost all time slots are used for transmitting either uncoded packets or coded (& unexpired) packets that are new/innovative to both users.

More rigorously, let $n(t)$ denote the value of the auxiliary variable n in the end of time slot t, which is the index of the next uncoded packet to be sent. Define the "index advancement" at time t as a function of t: $q(t) \stackrel{\Delta}{=} n(t) - \frac{t}{\lambda}$. Note that if $q(t)$ is bounded when $N \to \infty$, then we will have $t_N \approx \lambda N$ for large N. In the following, we assume $N = \infty$ (continuously streaming with no file-size limit) and use a Lyapunov function to prove that $q(t)$ is finite/stable for

[2] The oldest packet is the one with the smallest index.

$t \in [1, \infty)$ with probability one. Recall that

$$C_j(t) = \begin{cases} 1 & \text{if user } j \text{ receives the packet sent in slot } t \\ 0 & \text{otherwise} \end{cases}.$$

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 successfully arrive user j.

Following the protocol description in Section IV-A, we say that packet i is "a coding opportunity involving user j" if $i \in L_j$. That is, i being a coding opportunity involving j means that packet i has been received/decoded by the other user but not by user j. We then have the following central lemma.

Lemma 1: For any $t_1 < t_2$, if $t_2 < \lambda n(t_1)$, then

$$n(t_2) - n(t_1) \leq \max_{j=1,2} C_j(t_1, t_2) + 1.$$

Proof: Define $U^j_{t_1,t_2}$ as the number of time slots in $(t_1, t_2]$ when user j received an *uncoded* packet successfully; $M^j_{t_1,t_2}$ as the number of slots in $(t_1, t_2]$ when an *uncoded* packet is sent, user j does not receive it, but the other user received it successfully.

Since the n variable increment when and only when an uncoded packet is sent, and is received by at least one user (see Line IV-A of the pseudo-code), for any given user j we must have

$$n(t_2) - n(t_1) = U^j_{t_1,t_2} + M^j_{t_1,t_2} \tag{14}$$

Note that the uncoded packets received by the other user but not by j creates new coding opportunities involving user j. By construction, all these coding opportunities have index $\geq n(t_1)$. Further these new coding opportunities remain unexpired until time t_2 since $t_2 < \lambda n(t_1)$.

Define $V^j_{t_1,t_2}$ as the number of time slots when user j received a *coded* packet successfully during the interval $(t_1, t_2]$; $R^j_{t_2}$ as the number of coding opportunities remaining in the end of time slot t_2 involving user j. We then notice that each time slot when user j receives a coded packet successfully, user j can decode that packet, which thus destroys a coding opportunity involving user j. Hence,

$$V^j_{t_1,t_2} + R^j_{t_2} \geq M^j_{t_1,t_2} \tag{15}$$

The left-hand side of (15) is the number of coding opportunities destroyed due to successful decoding plus the number of remaining unexpired coding opportunities. The right-hand side of (15) is the number of coding opportunities created during the $(t_1, t_2]$ period. Since those coding opportunities do not expire during this period, these $M^j_{t_1,t_2}$ opportunities must either be destroyed during the period or remain in the end of time t_2. We thus have (15). The inequality sign is due to that before entering the $(t_1, t_2]$ period, there might already be some existing old coding opportunities at time t_1.

By definition, $U^j_{t_1,t_2} + V^j_{t_1,t_2} = C_j(t_1, t_2)$. Combining (14) and (15), we thus have

$$n(t_2) - n(t_1) \leq U^j_{t_1,t_2} + V^j_{t_1,t_2} + R^j_{t_2} \leq C_j(t_1, t_2) + R^j_{t_2}$$

Note that this inequality holds for both users. We also note that whenever there exists a coding opportunity ($|L_1| > 0$ and $|L_2| > 0$), the BS would send the coded packet, which destroys a coding opportunity upon the successful delivery. Only when at least one $|L_j| = 0$, the BS will send an uncoded packet, which increases L_j by at most one. From the above reasoning, at any time there exists at least one user whose number of coding opportunities is less than or equal to 1. Denote that user by j^*. We then have,

$$n(t_2) - n(t_1) \leq C_{j^*}(t_1, t_2) + 1 \leq \max_{j=1,2} C_j(t_1, t_2) + 1$$

The proof of Lemma 1 is complete. ■

We will use Lemma 1 to show that $q(t)$ stays finite with probability one. More explicitly, we will show that when $q(t) = B$ is sufficiently large, $\mathbb{E}\{q(t + \lambda B) - q(t) | q(t) = B\} < 0$. Namely, $q(t)$ has a negative drift when $q(t)$ is large. By using $q(t)$ itself as the Lyapunov function, $q(t)$ must be finite with probability one.

Suppose that $q(t_1) = n(t_1) - \frac{t_1}{\lambda} > B$. Choose $t_2 = t_1 + \lambda B$. By the law of large numbers, $C_j(t_1, t_2)$ concentrates around its expectation $\mathbb{E}\{C_j(t_1, t_2)\} = p(t_2 - t_1) = p\lambda B$ with high probability for sufficiently large B. Therefore, for any $\epsilon > 0$, and $\delta > 0$, we can find a B such that

$$\mathsf{P}\left(\frac{C_j(t_1, t_2)}{\lambda B} - p > \epsilon\right) < \delta, \text{ for } j = 1, 2.$$

So that

$$\begin{aligned} &\mathsf{P}\left(\frac{\max_{j=1,2} C_j(t_1, t_2)}{\lambda B} - p > \epsilon\right) \\ &\leq \mathsf{P}\left(\frac{C_1(t_1, t_2)}{\lambda B} - p > \epsilon\right) + \mathsf{P}\left(\frac{C_2(t_1, t_2)}{\lambda B} - p > \epsilon\right) \\ &\leq 2\delta \end{aligned}$$

By our construction of $t_2 = t_1 + \lambda B$ and its relationship to $q(t_1) > B$, we also have $t_2 < \lambda n(t_1)$. By Lemma 1, we have

$$n(t_2) - n(t_1) \leq \max_{j=1,2} C_j(t_1, t_2) + 1$$

And thus

$$\mathsf{P}\left(n(t_2) - n(t_1) \geq (p + \epsilon)\lambda B + 1 | q(t_1) > B\right) < 2\delta.$$

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

$$\begin{aligned} &\mathbb{E}\{n(t_2) - n(t_1) | q(t_1) > B\} \\ &\leq ((p + \epsilon)\lambda B + 1)(1 - 2\delta) + 2\lambda B\delta. \end{aligned}$$

By the above analysis, we have

$$\begin{aligned} &\mathbb{E}\left\{q(t_1 + \lambda B) - q(t_1) | \, q(t_1) > B\right\} \\ &= \mathbb{E}\left\{\left(n(t_2) - \frac{t_2}{\lambda}\right) - \left(n(t_1) - \frac{t_1}{\lambda}\right) \middle| \, q(t_1) > B\right\} \end{aligned}$$

$$\leq \left((p+\epsilon)(1-2\delta) - \frac{1}{\lambda} + 2\delta\right)\lambda B + 1 - 2\delta. \tag{16}$$

For any $p < \frac{1}{\lambda}$, one can choose sufficiently small ϵ and δ and a sufficiently large B such that the drift value in (16) is strictly negative. The negative drift can then be used to show that for any $\epsilon, \epsilon' > 0$, there exists an $N_0 > 0$ such that

$$\mathsf{P}\left(q((\lambda-\epsilon')N) \geq \frac{\epsilon' N}{\lambda}\right) < \epsilon, \tag{17}$$

for all $N > N_0$. For those $N > N_0$, we then have

$$\begin{aligned}
&\frac{\mathbb{E}\{N_{\text{success}}\}}{2N} \geq \frac{2p\mathbb{E}\{t_N\}}{2N} \\
&\geq \frac{2p(\lambda-\epsilon')N\mathsf{P}(t_N \geq (\lambda-\epsilon')N)}{2N} \\
&\geq \frac{2p(\lambda-\epsilon')N\mathsf{P}(q((\lambda-\epsilon')N) + \frac{(\lambda-\epsilon')N}{\lambda} < N)}{2N} \\
&= p(\lambda-\epsilon')(1 - \mathsf{P}(q((\lambda-\epsilon')N) \geq \frac{\epsilon' N}{\lambda})) \\
&\geq p(\lambda-\epsilon')(1-\epsilon),
\end{aligned} \tag{18}$$

where the first inequality follows from focusing on the expected rewards obtained during the random period $[1, t_N]$ and from the optional sampling theorem for Martingales. The second inequality follows from the Markov inequality. The third inequality follows from the fact that when $n((\lambda-\epsilon')N) < N$ we must have $t_N \geq (\lambda-\epsilon')N$. By letting $N \to \infty$ and then $\epsilon, \epsilon' \to 0$, the asymptotic throughput is $\geq \lambda p$. The proof for the case $p < \frac{1}{\lambda}$ is complete.

Now we consider the case when $p > \frac{1}{\lambda}$. Let $\lambda' = \frac{1}{p+\overline{\epsilon}} < \lambda$, where $\overline{\epsilon} > 0$ is a small number. We would like to have $p + \overline{\epsilon}$ being a fractional number so that we can choose a sufficiently large B such that $B\lambda'$ being an integer. Note 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 $q'(t) > B$. Suppose $q'(t_1) = n(t_1) - \frac{t_1}{\lambda'} > B$ for some t_1. Set $t_2 = t_1 + \lambda' B$. Therefore, for any $\epsilon > 0$, and $\delta > 0$, we can find a large enough B such that

$$\mathsf{P}\left(\frac{C_j(t_1,t_2)}{\lambda' B} - p > \epsilon\right) < \delta, \text{ for } j = 1, 2.$$

Hence, we have

$$\begin{aligned}
&\mathsf{P}\left(\frac{\max_{j=1,2} C_j(t_1,t_2)}{\lambda' B} - p > \epsilon\right) \\
&\leq \mathsf{P}\left(\frac{C_1(t_1,t_2)}{\lambda' B} - p > \epsilon\right) + \mathsf{P}\left(\frac{C_2(t_1,t_2)}{\lambda' B} - p > \epsilon\right) \\
&\leq 2\delta.
\end{aligned}$$

By our construction of $t_2 = t_1 + \lambda' B$ and its relationship to $q'(t_1) > B$, we also have $t_2 < \lambda' n(t_1) \leq \lambda n(t_1)$. By Lemma 1, we have

$$n(t_2) - n(t_1) \leq \max_{j=1,2} C_j(t_1,t_2) + 1,$$

and

$$\mathsf{P}\left(n(t_2) - n(t_1) \geq (p+\epsilon)(\lambda' B) + 1\right) < 2\delta.$$

Since we also have $n(t_2) - n(t_1) \leq t_2 - t_1 = \lambda' B$. Jointly, we have

$$\begin{aligned}
&\mathbb{E}\{n(t_2) - n(t_1) | q'(t_1) > B)\} \\
&\leq ((p+\epsilon)\lambda' B + 1)(1-2\delta) + 2\lambda' B\delta.
\end{aligned}$$

By the above analysis, the conditional expectation that $q(t_1) > B$ must satisfy

$$\begin{aligned}
&\mathbb{E}\left\{q'(t_1 + \lambda' B) - q'(t_1) \middle| q'(t_1) > B\right\} \\
&= \mathbb{E}\left\{\left(n(t_2) - \frac{t_2}{\lambda'}\right) - \left(n(t_1) - \frac{t_1}{\lambda'}\right) \middle| q'(t_1) > B\right\} \\
&\leq \left((p+\epsilon)(1-2\delta) - \frac{1}{\lambda'} + 2\delta\right)(t_2 - t_1) + 1 - 2\delta. \\
&= \left((p+\epsilon)(1-2\delta) - \frac{1}{\lambda'} + 2\delta\right)\lambda' B + 1 - 2\delta.
\end{aligned} \tag{19}$$

Since $p < \frac{1}{\lambda'}$, one can choose sufficiently small ϵ and δ and a sufficiently large B such that the drift in (19) is strictly negative. The negative drift can then be used to show that for any $\epsilon, \epsilon' > 0$, there exists an $N_0 > 0$ such that

$$\mathsf{P}\left(q((\lambda'-\epsilon')N) \geq \frac{\epsilon' N}{\lambda'}\right) < \epsilon, \tag{20}$$

for all $N > N_0$. For those $N > N_0$, we then have

$$\begin{aligned}
&\frac{\mathbb{E}\{N_{\text{success}}\}}{2N} \geq \frac{2p\mathbb{E}\{t_N\}}{2N} \\
&\geq \frac{2p(\lambda'-\epsilon')N\mathsf{P}(t_N \geq (\lambda'-\epsilon')N)}{2N} \\
&\geq \frac{2p(\lambda'-\epsilon')N\mathsf{P}(q((\lambda'-\epsilon')N) + \frac{(\lambda'-\epsilon')N}{\lambda'} < N)}{2N} \\
&= p(\lambda'-\epsilon')(1 - \mathsf{P}(q((\lambda'-\epsilon')N) \geq \frac{\epsilon' N}{\lambda'})) \\
&\geq p(\lambda'-\epsilon')(1-\epsilon),
\end{aligned} \tag{21}$$

where the first inequality follows from focusing on the expected rewards obtained during the random period $[1, t_N]$ and from the optional sampling theorem for Martingales. The second inequality follows from the Markov inequality. The third inequality follows from the fact that when $n((\lambda'-\epsilon')N) < N$ we must have $t_N \geq (\lambda'-\epsilon')N$. By letting $N \to \infty$ and then $\epsilon, \epsilon' \to 0$, the asymptotic throughput is $\geq p\lambda'$. When $\overline{\epsilon} \to 0$, so that $p \to \frac{1}{\lambda'}$, the asymptotic throughput can achieve the capacity upper bound. The proof for the case $p \geq \frac{1}{\lambda}$ is complete.

The finiteness of the index advancement $q(t)$ is also highly relevant to the *transmission delay* in the setting of *sequential packet arrivals*. Our proof can be used to show that the total transmission delay (from the packet arrival at the base station to decoding at the individual user) remains finite with probability one. We will discuss this point in more detail in Section V-C.

In our scheme, the BS sends the uncoded packets first and combines two information packets together only when there are coding opportunities involving both users. Namely, we only transmit network coded packet that can serve both users

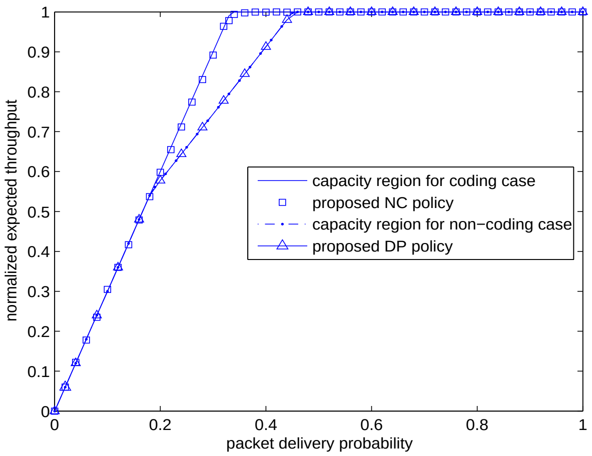

Fig. 2. The capacity curves for the dynamic-programming non-coding policy and for the proposed network coding policy.

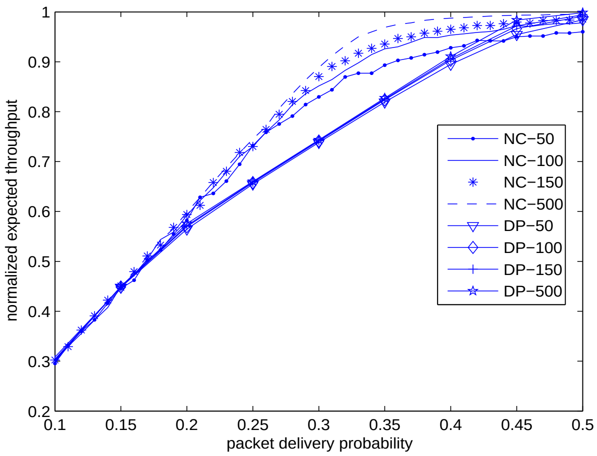

Fig. 3. Performance comparison between NC and DP policies for small finite $N = 50,\ 100,\ 150,\ 500$.

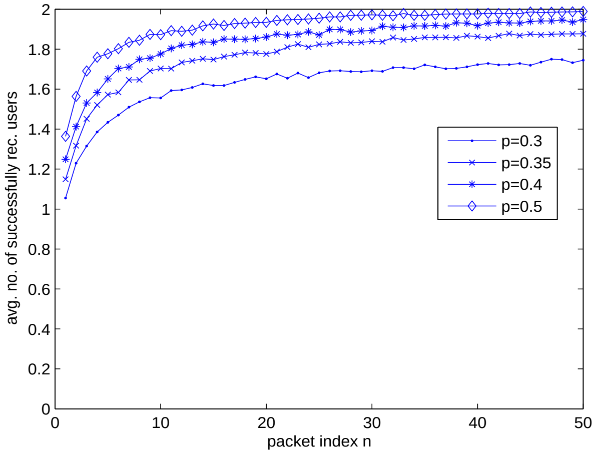

Fig. 4. The packet-by-packet delivery rate of the NC scheme for first 50 packets. The deadline is $3n$ for the n-th packet.

simultaneously. It is possible that when we are waiting for such beneficial coding opportunities, some of the packets that have only been received by one user may have expired. Since those packets are heard by only one user, the throughput suffers. However, the above proof shows that the proposed scheme is asymptotically throughput optimal, which means that such *expiration of coding opportunities* happens only infrequently. Most of the packets can be pumped through the broadcast channel within the deadline constraints even if we wait for the occurence of the coding opportunities that benefit both users.

V. Simulation

Our previous analyses focus on the asymptotic case when $N \to \infty$. In this section, we use simulation to verify the performance of the uncoded DP policy and the proposed network coding (NC) scheme for finite N. For all our simulation results, we assume that the deadline of the n-th packet is $3n$, i.e., $\lambda = 3$.

A. Performance for Large N

Fig. 2 contains four curves: the asymptotic capacity regions for coded and uncoded transmissions and the achievable throughputs for the proposed NC scheme and the DP non-coding policy with $N = 10000$. The capacity regions are plotted according to the closed form expressions in Sections III-B and IV-B. For the DP non-coding policy, for each given p value, we evaluate the expected throughput by the iterative equations in (2). For the proposed NC policy, for each given p value, we run the simulation and count the number of successes for both users.

As illustrated in Fig. 2, the asymptotic capacity curves for coded/uncoded transmissions fit the numerical results for large $N = 10000$. When p value is less than $\frac{3-\sqrt{3^2-3}}{3} \approx 0.18$ or when $p > \frac{3+1-\sqrt{3^2+1-3}}{3} \approx 0.45$, both the DP policy and the NC policy achieve the broadcast channel capacity. However, when $0.18 < p < 0.45$, the performance of the best non-coding policy is strictly worse than the broadcast capacity, which can only be achieved by the NC policy. The largest performance gain happens at the critical erasure probability $p = \frac{1}{\lambda} = 1/3$, for which we see 25% throughput improvement for the NC policy over the DP policy.

B. Performance for Small N

In Fig. 3 we compare the NC and DP policies for small, finite file size N. Even for file size as small as $N = 50$, the performance of NC scheme is better than that of DP. We also note that the expected rewards for small N follow closely with their asymptotic counterparts, especially when the DP policy is used. The normalized expected rewards for the DP policy is almost indistinguishable from $N = 50$ to 500. The expected throughput of the NC scheme deviates slightly more from its asymptotic expression for small N (i.e., $N < 500$). The performance degradation of the NC 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. In Fig. 4 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. 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. 4. The relatively large packet loss for the initial packets (those with small n) is the cause of the throughput degradation in Fig. 3. 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. 3 by the intersecting point of the "NC-50" curve and the $p = 0.5$ vertical line.

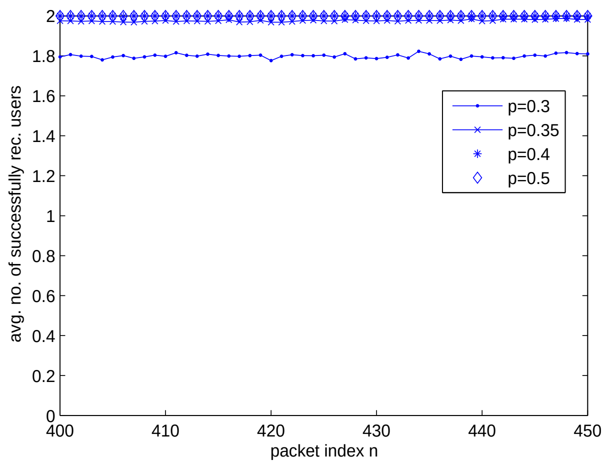

Fig. 5. The packet-by-packet delivery rate of the NC scheme for packets with indicies between 400 and 450. The deadline is $3n$ for the n-th packet

We also plot the average number of users that receive the n-th packet before the deadline for $n = 400$ to 450 in Fig 5. 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-C. As illustrated in Fig 5, the per-packet throughput for $p = 0.3$ approaches the upper bound $1.8 = 0.9 \times 2$ users for the packets with indicies between 400 and 450. This also verifies the asymptotic optimality of the proposed scheme.

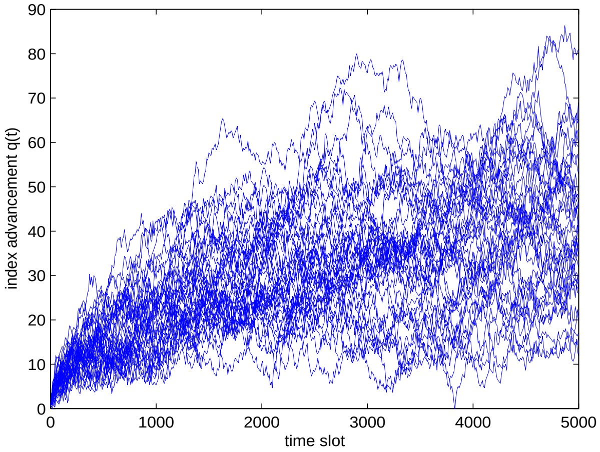

Fig. 6. Time evolution of the index advancement $q(t)$ for $p = 0.33$.

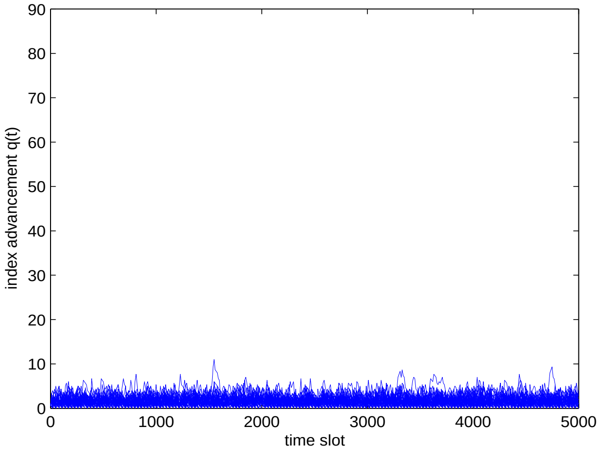

Fig. 7. Time evolution of the index advancement $q(t) = n(t) - \frac{t}{\lambda}$ for $p = 0.25$.

On the other hand, 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-C and verified in Fig. 2.

C. *Time Evolution of* $q(t)$

Fig. 6 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-C, $q(t)$ 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 (16), 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. 7) 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-C 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.

VI. Conclusions

In this work, we have modeled and analyzed the streaming broadcast problem over downlinks in a single cell. We have characterized the optimal throughput for both the uncoded and coded transmission schemes with hard deadline constraints, and shown that the capacity region of network coded transmission is strictly larger than that of the uncoded schemes.

In the uncoded case, an optimal policy has been devised based on finite-horizon dynamic programming (DP) within a Markov decision framework. And we have obtained a closed form expression of the optimal throughput in the asymptotic sense, i.e., when broadcasting files of size $N \to \infty$. For coded transmission, we have proposed a novel network coding (NC) scheme, which achieves the asymptotic capacity without the knowledge about the packet delivery probability. In addition to the asymptotic analysis, our simulation results show that the proposed NC scheme achieves strictly higher throughput than that of the best uncoded scheme even for very small file size N. A new Lyapunov analysis of the *index advancement* has been developed, which sheds further insight into the NC scheme for both the file-streaming and live-streaming applications.

For the uncoded case, the analysis techniques in this work can be immediately applied to the case of more than two users. We are currently investigating the Lyapunov analysis for $>$ 2 users and its corresponding applications for the setting of sequential arrivals and even with *random arrival processes*.

References

[1] C.-J. Su and L. Tassiulas, "Broadcast scheduling for information distribution," in *16th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, April 1997, pp. 109–117.

[2] S. Jiang and N. Vaidya, "Scheduling data broadcast to 'impatient' users," in *1st ACM international workshop on Data engineering for wireless and mobile access*, 1999, pp. 52–59.

[3] D. Aksoy and M. Franklin, "Scheduling for large-scale on-demand data broadcasting," in *17th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 29–April 2 1998, pp. 651–659.

[4] 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.

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

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

[7] 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.

[8] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. of 41st Annual Allerton Conference on Communication, Control, and Computing*, October 2003, pp. 40–49.

[9] J. Sundararajan, D. Shah, and M. Médard, "ARQ for network coding," in *Proc. of IEEE International Symposium on Information Theory*, July 2008, pp. 1651–1655.

[10] 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 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, September 2006, pp. 243–254.

[11] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," in *ACM International Conference on Mobile Computing and Networking (MobiCom)*, September 2007, pp. 135–146.

[12] E. Erez, M. Efros, and T. Ho, "Network codes with deadlines," in *Proc. of 46th Annual Allerton Conference on Communication, Control, and Computing*, September 2008, pp. 339–346.

[13] A. Eryilmaz, A. Ozdaglar, and M. Médard, "On delay performance gains from network coding," in *Proc. of 40th Annual Conference on Information Sciences and Systems*, March 2006, pp. 864–870.

[14] J. Barros, R. Costa, D. Munaretto, and J. Widmer, "Effective delay control in online network coding," in *the 28th Conference on Computer Communications (INFOCOM)*, April 2009, pp. 208–216.

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

[16] W. Yeow, A. Hoang, and C. Tham, "Minimizing delay for multicast-streaming in wireless networks with network coding," in *the 28th Conference on Computer Communications (INFOCOM)*, April 2009, pp. 190–198.

[17] V. Raghunathan, V. Borkar, M. Cao, and P. Kumar, "Index policies for real-time multicast scheduling for wireless broadcast systems," in *the 27th Conference on Computer Communications (INFOCOM)*, April 2008, pp. 1570–1578.

[18] D. Nguyen, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," in *NetCod 2007. Fourth Workshop on Network Coding, Theory and Applications*, January 2007.