7-23-2012

Inter-Session Network Coding Schemes for Two Unicast Sessions with Sequential Hard Deadline Constraints

Xiaohang Li
Purdue University, li179@purdue.edu

Chih-Chun Wang
Purdue University, chihw@purdue.edu

Xiaojun Lin
Purdue University, linx@ecn.purdue.edu

Inter-Session Network Coding Schemes for Two Unicast Sessions with Sequential Hard Deadline Contraints

Xiaohang Li

Chih-Chun Wang

Xiaojun Lin

TR-ECE-12-06

July 23, 2012

School of Electrical and Computer Engineering

1285 Electrical Engineering Building

Purdue University

West Lafayette, IN 47907-1285

Inter-Session Network Coding Schemes for Two Unicast Sessions with Sequential 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 47907
Email: {li179, chihw}@purdue.edu, linx@ecn.purdue.edu

***Abstract*—The emerging wireless media delivery services have placed greater demands for wireless networks to support high-throughput applications while minimizing the delay of individual packets. In this paper, we investigate using inter-session network coding to send packets wirelessly for two *deadline-constrained* unicast sessions. Specifically, each unicast session aims to transmit a stored video file, whose packets have hard sequential deadline constraints. We first characterize the corresponding deadline-constrained capacity region under heterogeneous channel conditions and heterogeneous deadline constraints. We show that this deadline-constrained capacity region can be achieved *asymptotically* by modifying the existing *generation-based schemes*. Despite its asymptotic optimality, the generation-based scheme has poor performance and high complexity in the practical regime small & medium file sizes. To address these problems, we further develop new immediately-decodable network coding (IDNC) schemes that admit superior performance in the practical regime while being provably optimal in the asymptotic regime. In contrast to the existing delay/deadline-based IDNC results, which focus on a single multicast session (intra-session network coding) with homogeneous channel conditions, our new IDNC design takes full account of channel heterogeneity and provides the first rigorous asymptotic optimality analysis for two unicasts with (potentially heterogeneous) hard deadline constraints.**

I. Introduction

The advance of broadband wireless technologies has enabled a number of innovative wireless services. It is now common to use 3G/4G cellular networks or WiFi to provide multimedia services, most of which have stringent Quality-of-Service (QoS) requirements. Among them, video streaming over wireless networks has gained a significant amount of interest. For such multimedia traffic, unicast is the prevalent mode of operation since different users often request different contents. In this paper, we consider sending two unicast sessions over an unreliable wireless channel. Each unicast session downloads a stored-video file from the base-station (BS). 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 the 1/30 second, while the second frame's deadline is at the 2/30 second, and so on). If a packet is not delivered before the deadline, it is considered useless to the receiver. 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. Meanwhile, the asymmetry due to heterogeneous channel conditions and heterogeneous deadlines imposes further difficulties for jointly scheduling multiple deadline-constrained unicast sessions. In this paper, we are interested in using inter-session network coding (NC) to improve the deadline-constrained streaming throughput in this setting.

It is well-known that without deadline constraints, NC can increase the throughput of communication networks [1], [2] while still admitting efficient implementation [3], [4]. While it has been shown that NC is particularly attractive for wireless broadcast in our prior work [5], [6], it is notable that NC can also improve the throughput for multiple unicast sessions as well [7]. However, if not properly designed, NC could introduce "decoding delay," i.e., the receiver may not be able to decode the information packet right away. For example, in the *generation-based NC schemes* [4], each user 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 subject to the deadline constraints becomes a challenging problem.

Existing studies have discussed different aspects of inter-session NC transmission schemes. However, they either do not account for the lossy wireless network setting, or do not consider the delay aspect. Specifically, [8]–[10] discuss how to design and control intersession-network-coded traffic for the setting of lossless channels. [7] proposes a practical network coding scheme for multiple unicast-sessions while [11], [12] characterize the corresponding information-theoretic capacity region. [13] combines intra- and inter-session network coding to enhance the throughput of unicast flows. Recently, [14] characterizes the capacity of 2-session unicast for an access-point network. These studies focus on throughput without considering delay. In contrast, our paper focuses on the delay aspect when coding over two unicast sessions. Readers are referred to [6], [15]–[19] and the references therein for the delay analysis in the simpler[1] setting of a single multicast/broadcast session.

In this work, we first modify the generation based (GB) scheme to achieve the hard-deadline-constrained capacity asymptotically. We then show the bad performance of the start-up phase for GB scheme. Further, we analyze the delay inefficiency that causes GB scheme to perform poorly in the practical regime of median file sizes. To combat the delay

This work has been partially supported by the NSF grants CNS-0721484, CNS-0721477, CNS-0643145, CCF-0845968, CNS-0905331, and a grant from Purdue Research Foundation. Part of this work has appeared in Allerton Conference 2011 as an invited paper.

[1]It is well known [14] that even without the delay consideration, the capacity / throughput study of coding over multiple unicast sessions is much more challenging than that of coding over a single multicast session.

inefficiency of most existing NC schemes, recent practical protocols have focused more on the "immediately decodable" NC (IDNC) schemes [5]–[7], [20]. In this work, we are interested in developing new IDNC schemes to maximize the throughput for each unicast session under the sequential deadline constraints of stored-video streaming. Unfortunately, the performance analysis of these IDNC schemes turns out to be highly non-trivial. In contrast to our prior work [5], [6] that focus on a single multicast session with homogeneous channel conditions and deadline constraints, the design and performance analysis of the IDNC scheme is much more complicated for unicast-sessions because of the asymmetry due to heterogeneous channel conditions and heterogeneous deadline constraints (see further discussions in Section V). Nonetheless, we establish the asymptotic optimality of the proposed IDNC scheme when the file sizes are large. In this analysis, we use a novel form of Lyapunov function, which reveals new and intricate dynamics of an IDNC system. Further, our numerical simulations show that the throughput of the IDNC scheme is close-to-optimal even for small file sizes. We believe that our study on the 2-user case uncovers non-trivial and interesting insights that could serve as a precursor to the full design and analysis for the case of a larger number of users. Prior studies of similar IDNC schemes either do not consider deadline-constraints at all [21], or only consider the multicast case [6]. 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 in the multi-unicast setting.

The rest of this paper is organized as follows. Section II introduces the system model. Section III discusses the capacity region with deadline constraints. Section IV introduces the generation based scheme for sequential hard deadline constraints. Section V describes the IDNC schemes for deadline-constrained streaming. Section VI provides the throughput analysis of IDNC schemes under heterogeneous deadline constraints and heterogeneous channel conditions, which is the main contribution of this paper. Section VII presents the simulation results for the proposed IDNC schemes. Section VIII concludes the paper.

II. The Setting

We consider the scenario that the base station (BS) sends two video files to 2 users, d_1 and d_2, respectively. The two video files contain N_1 and N_2 packets, respectively, and are denoted by $\{X_{1,n}\}_{n=1}^{N_1}$, $\{X_{2,n}\}_{n=1}^{N_2}$, respectively. We sometimes use session 1 and session 2 to refer to (the transmission of) the data packets for d_1 and d_2, respectively.

We define the time when the BS starts transmission as the time origin, and assume that all packets are available at the BS at time 0. We assume slotted transmission. Each packet $X_{j,n}$ ($j = 1, 2$) has a deadline $\tau_{j,n}$ such that after time slot $\tau_{j,n}$ the packet $X_{j,n}$ is no longer useful for user j. We assume that for $j = 1, 2$

$$\tau_{j,n} = \lambda_j \cdot n, \quad n \in \{1, \ldots, N_j\}, \tag{1}$$

where λ_j is the (sequential) deadline increment for session j. In this work, we consider heterogeneous deadlines, i.e., λ_1 and λ_2 may be different. We assume that $T = \lambda_1 N_1 = \lambda_2 N_2$, that is, the total display time T for each video file is the same[2].

We consider random and unreliable wireless channels. Both users can overhear the transmission with certain probability. For $j = 1, 2$, we use $C_j(t) = 1$ to denote the event that user j can receive a packet successfully at time t; and $C_j(t) = 0$, otherwise. In this work, we assume channels are independently and identically distributed (i.i.d.) across time, and $C_1(t)$ and $C_2(t)$ are independent with each other. The success probabilities for channels 1 and 2 are denoted by p_1 and p_2, respectively. We consider heterogeneous channels, i.e., p_1 may be different from p_2. We assume that both p_1 and p_2 are known to the BS. We also assume that at the end of each time slot, the BS has perfect feedback from both users regarding whether the transmitted packet has been successfully received by each user. In one slot, the BS can code a set of unexpired packets together and send the resultant coded packet to all users. When coding is used, we say that the original packet is correctly received only if it can be "decoded" from the coded transmission before the corresponding deadline.

Our goal is to design a coding/scheduling policy that maximizes the number of successful (unexpired) packet receptions. More specifically, let $D_j(n) = 1$ if user j can successfully decode/recover $X_{j,n}$ before its deadline $\tau_{j,n}$; and $D_j(n) = 0$, otherwise. We define the total number of unexpired successes by $N_1^{\text{success}} \triangleq \sum_{n=1}^{N_1} D_1(n)$ and $N_2^{\text{success}} \triangleq \sum_{n=1}^{N_2} D_2(n)$. Our goal is to maximize the minimum of the normalized throughputs, between the two users, i.e., maximizing $\min\left(\frac{\mathsf{E}\{N_1^{\text{success}}\}}{N_1}, \frac{\mathsf{E}\{N_2^{\text{success}}\}}{N_2}\right)$.

III. The Deadline-Constrained Capacity Region

Consider an interval $(0, T]$. Suppose that during this interval, on average $r_1 T$ packets from session 1 can be delivered before their deadlines, where r_1 is termed the achievable rate for user 1. Obviously, $r_1 \leq \frac{1}{\lambda_1}$ since the best scenario is to deliver all N_1 packets before $T = \lambda_1 N_1$. Similarly, suppose on average $r_2 T$ packets from session 2 can also be delivered in this period, where $r_2 \leq \frac{1}{\lambda_2}$ is the achievable rate for user 2. In [14] and [22], it is shown that even when not considering the sequential deadline constraints, the best possible achievable rate pairs (r_1, r_2) must satisfy the following two inequalities simultaneously:[3]

$$\frac{r_1}{p_1} + \frac{r_2}{1 - (1 - p_1)(1 - p_2)} \leq 1 \tag{2}$$

[2]If the display time of one file is longer than that of the other, then after the completion time of the other file (before which both files were inter-session coded) we can treat the remaining packets as a single, separate unicast session, which is much easier to deal with, since there is no other session to be coded together.

[3]The intuition behind these two inequalities are as follows. Consider (2) first. Since we would like to send $r_1 T$ packets to d_1, transmitting those packets (either in an uncoded or in a coded way) would require $\frac{r_1 T}{p_1}$ number of time slots on average. Note that even though sometimes we may use NC to serve two destinations simultaneously, roughly speaking before doing so some version of each session-2 packet needs to be received by at least one of the destinations before it can be mixed with a session-1 packet [14], [22]. As a result at least $\frac{r_2 T}{1-(1-p_1)(1-p_2)}$ number of time slots should be dedicated to sending session-2 packets (not mixing with any session-1 transmission). Since the total time budget is T, the above heuristics imply (2). By swapping the roles of sessions 1 and 2, we also have (3).

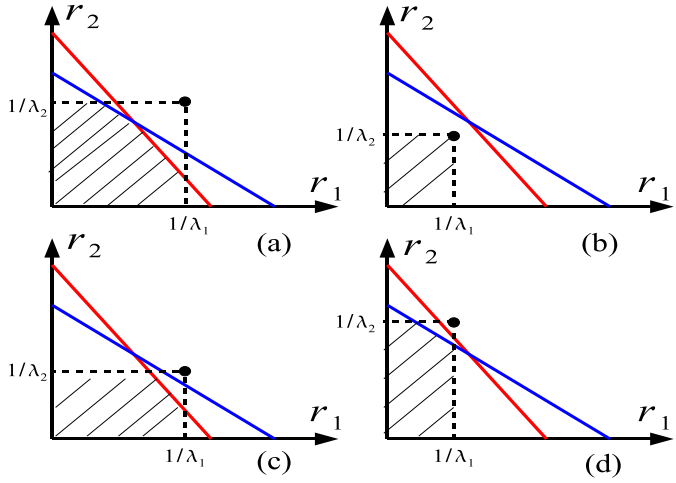

Fig. 1. Asymptotic capacity region for a hard-deadline-constrained two unicast system. Subfigures (a) to (d) represent four possible cases depending on the location of the point $(\frac{1}{\lambda_1}, \frac{1}{\lambda_2})$ and the red and the blue line segments.

$$\frac{r_2}{p_2} + \frac{r_1}{1-(1-p_1)(1-p_2)} \leq 1. \tag{3}$$

Since the capacity without deadlines is always an upper bound of the capacity with deadlines, the above analysis proves the following outer bound on the deadline-constrained capacity.

Proposition 1. *For any scheme in a deadline constrained system, the achievable throughput vector* $\left(\frac{\mathsf{E}\{N_1^{success}\}}{\lambda_1 N_1}, \frac{\mathsf{E}\{N_2^{success}\}}{\lambda_2 N_2}\right)$ *must be in the following region:*

$$\mathcal{R} = \Big\{(r_1, r_2) : 0 \leq r_1 \leq \frac{1}{\lambda_1}, 0 \leq r_2 \leq \frac{1}{\lambda_2}, \text{ and } (r_1, r_2) \text{ satisfies (2) and (3) simultaneously}\Big\}. \tag{4}$$

We will prove later that for sufficiently large T, the above capacity outer bound can be achieved by either a generation-based scheme or an IDNC scheme. The region in (2) and (3) thus describes the asymptotic capacity region for a deadline-constrained system.

We illustrate the capacity region in Fig 1. The red and blue lines represent the constraints by (2) and (3), respectively. The shadowed area indicates the asymptotic capacity region depending on the relative location of the point $(\frac{1}{\lambda_1}, \frac{1}{\lambda_2})$ and the red and blue line segments.

By similar analysis, we can also prove that if coding is prohibited, then the non-coding capacity region of a deadline-constrained system becomes

$$\mathcal{R}^{\text{uncoded}} = \Big\{(r_1, r_2) : 0 \leq r_1 \leq \frac{1}{\lambda_1}, 0 \leq r_2 \leq \frac{1}{\lambda_2}, \text{ and } (r_1, r_2) \text{ satisfies } \frac{r_1}{p_1} + \frac{r_2}{p_2} \leq 1\Big\}. \tag{5}$$

IV. Achieving The Asymptotic Capacity by A Generation-Based Scheme

The generation-based (GB) scheme is widely used in existing work [4], [14] for throughout-oriented analysis. Specifically, the GB scheme divides the whole file into several generations and transmits each generation sequentially. Within each generation, the BS encodes all the packets that belong to this generation together and transmits the coded packets. After receiving enough coded packets, the receiver can decode the entire generation. The BS then moves on to the next generation. Since the receiver needs to collect enough packets before decoding, a GB scheme generally incurs a decoding delay (the larger the generation size, the longer the decoding delay). For the following, we will show that the GB scheme in [14] can be modified to achieve the asymptotic capacity in Proposition 1, and then elaborate on its problem in the practical regime of median file sizes.

Specifically, for sessions 1 and 2 we choose the corresponding generation sizes to be M_1 and M_2, respectively, and we enforce that $\lambda_1 M_1 = \lambda_2 M_2$. (For practical implementation, we can relax this requirement.) In this way, both sessions will have the same number of generations. The l-th generation of session-1 packets can be coded together with the l-th generation of session-2 packets. We then note that the GB scheme proposed in [14] cannot be used directly in a deadline-constrained system due to the following two observations.

First, recall that our goal is to send all N_1 and N_2 packets (before their deadlines) within the interval $(0, T]$ where $T = \lambda_1 N_1 = \lambda_2 N_2$. Therefore, the best scenario is to sustain the rate $(1/\lambda_1, 1/\lambda_2)$. However, $(1/\lambda_1, 1/\lambda_2)$ may be outside the deadline-constrained capacity outer-bound in Proposition 1, also see Fig. 1(a,c,d). In this case, we say that the system is *under-provisioned* [23]. The problem for an under-provisioned system is that it is simply impossible for every packet to meet its deadline constraint. However, a GB scheme will encode all packets of the same generation together and decode all the packets together. Therefore, if there is any packet that cannot meet its deadline constraint, then the entire generation cannot be decoded, which greatly reduces the throughput. Our solution to this problem is to deliberately discard some packets so that those packets do not participate in the GB scheme. In this way, those not-discarded packets have a better chance to be decoded in a GB scheme. To facilitate the exhibition, we modify the generation based scheme proposed in [14] to fit the sequential hard deadline constraints. The new generation based scheme is different from the one in [14] from two aspects: First, suppose that the best possible scenario (in which all packets can be successfully decoded in time) is simply not sustainable by the underlying channel quality (p_1, p_2). Namely, when the rate pair $(1/\lambda_1, 1/\lambda_2)$ violates either (2) or (3), it is simply impossible to meet the deadlines of all packets. Recall that this is the *under-provisioned* scenario. By deliberately discarding some packets we relax the deadlines for those not-discarded packets. Therefore, those not-discarded packets are less likely to expire. So we also incorporate the dropping mechanism for generation based scheme for the under provisioned case since otherwise decoding would be extremely difficult.

Second, there is little time to perform coding for the first few packet since the first few packets expire very quickly. Our solution to this issue is to drop the first generation of both session 1 and session 2, and start encoding generation-2 packets from the very beginning. In this way, we allow more time for all the subsequent encoding/decoding.

After considering these two points, we can design a Generation-Based scheme for the hard deadline constraints as follows. For simplicity, we use γ to denote a constant value used throughout the algorithm, which can be easily computed by the BS. That is,

$$\gamma \triangleq \min\Big(\frac{1}{\frac{1/\lambda_1}{p_1}+\frac{1/\lambda_2}{p_1+p_2-p_1p_2}}, \frac{1}{\frac{1/\lambda_1}{p_1+p_2-p_1p_2}+\frac{1/\lambda_2}{p_2}}\Big). \quad (6)$$

§ GENERATION-BASED-SCHEME

1: Drop the first generation of both session 1 and session 2. Set GenID$\leftarrow 2$
2: For $j = 1, 2$ choose arbitrarily $M_j(1 - \min(\gamma, 1))$ user-j packets from the GenID-th generation and *drop those packets*, i.e, we remove those packets from any future consideration.
3: **for** time $t = (\text{GenID} - 2) * \lambda_1 M_1 + 1$ to time $t = (\text{GenID} - 1) * \lambda_1 M_1$ **do**
4: **if** there is still a user-j packet of the GenID-th generation that is not heard by any user **then**
5: The BS transmits one of such packets (that is not heard by any user) uncodedly.
6: **else**
7: After all GenID-th packets have been heard by at least one user, using the idea of Random Linear Network Coding [3], the BS generates a single coded packets by randomly mixing all user-1 packets in the GenID-th generation that have been heard only by user 2, and all user-2 packets that have been heard only by user 1. The BS sends the RLNC-generated packet.
8: **end if**
9: **end for**
10: In the end of time $t = (\text{GenID} - 1) * \lambda_1 M_1$, user 1 (resp. user 2) will decode if it has received enough coded packets of the GenID-th generation.
11: GenID$\leftarrow$ GenID+1 and go back to Line 2.

It is easy to see that the generation based scheme is throughput optimal in the asymptotic sense (i.e., when the generation size is sufficiently large, and when the file sizes N_1 and N_2 approach infinity [4]). To see this, consider first the over-provision case. Suppose the size of each generation is large enough. Since $\gamma > 1$, by the law of large numbers, we can repeat the analysis in Section III. Then, we have, for close-to-1 probability, each generation can be transmitted successfully to users for each session. If the number of generations also approaches infinity, the loss due to the dropping of the first generation can be neglected. Thus, the asymptotic throughput optimality for the over-provisioned case can be established. Next, consider the under-provisioned case. After dropping a certain number of packets, the system is able to accommodate the transmission of the remaining packets for each generation. Thus we can also show the asymptotic throughput optimality for the under-provisioned case. There is a problem with the generation based scheme, however. Note that the larger the generation size is, intuitively the better the throughput for generation based scheme. However, in practice, if the size of each generation is large, then the performance for the start-up phase may be poor. Because the first generation is dropped, the larger the generation size is, the poorer the performance in the initial period. On the other hand, if the generation size is small, then the law of large numbers cannot kick in. There will be a large chance that, an insufficient number of coded packets for generation j are received before time $(j-1)M_1\lambda_1$ (after this generation $j+1$ will start). If this happens, all the coded packets have to be dropped and cannot be decoded. Thus, the throughput will suffer.

These insights can be verified through our simulation results comparing the performance for both cases with the small file size and large file size. Here we use "G-B 4-4" in short of generation based scheme with generation size 4 and 4, respectively, for session 1 and session 2. "G-B 40-40" denotes generation based scheme with generation size 40 and 40, respectively. "IDNC" denotes the IDNC scheme that we would discuss in Section V. "upper bound" denotes the upper bound derived from Proposition 1, while "upper bound for uncoded" denotes the upper bound derived from (5). We set $\lambda_1 = 3$, $\lambda_2 = 3$, and $p_1 = p_2$. In Fig. 2 we compare the performance for large file sizes, and we set $N_1 = 40000$, $N_2 = 40000$. We can see that, G-B 40-40 performs better than G-B 4-4, since larger generation size can bring higher throughput. In Fig. 3 we compare the performance for small file sizes, and we set $N_1 = 400$, $N_2 = 400$. We can see that most of time G-B 4-4 suffers less in throughput compared with G-B 40-40, as G-B 4-4 drops less packets in the beginning.

As can be seen in the figures, although the GB schemes are asymptotically optimal, they have poor performance in a practical regime of median file sizes. Most of the time, their performance is barely better than the non-coded solution (as compared with the upper bound for non-coded schemes). Further, the GB scheme also suffers from high decoding complexity when a large generation size is used. Buffer management is also an issue in a GB scheme since the users need to store all the received coded packets before decoding in the end. In the remaining sections of this paper, we propose a new Immediately Decodable Network Coding (IDNC) scheme that addresses the above issues, which has superior performance at median file sizes, and is also provably optimal in the asymptotic regime.

V. THE IDNC SCHEME

To overcome the delay inefficiency of generation based scheme, recent practical protocols have focused more on the "immediately decodable" NC (IDNC) schemes [7], [20]. An IDNC scheme for two unicast sessions has the following structure. Suppose that two users d_1 and d_2 are interested in different packets X and Y, respectively. Initially, the BS sends X and Y uncodedly until each packet is received by at least one user. Suppose due to random channel realization, d_1 has overheard Y and d_2 has overheard X. We call the (unexpired) packet X a (potential) coding opportunity involving user 1 and call the (unexpired) packet Y a (potential) coding opportunity

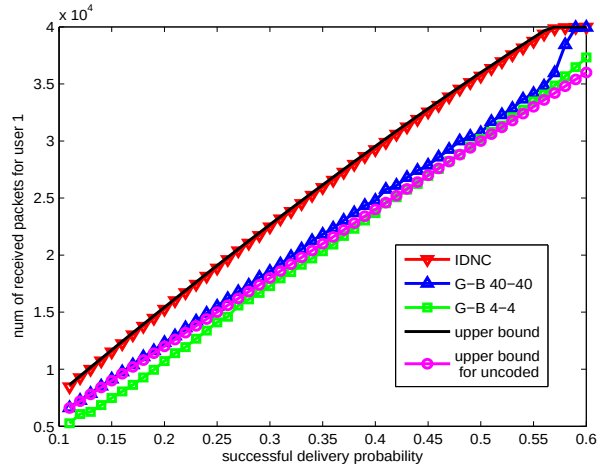

Fig. 2. Total number of received packets in session 1 when $\lambda_1 = 3$, $\lambda_2 = 3$, $N_1 = 40000$, $N_2 = 40000$ averaged in 10 simulations.

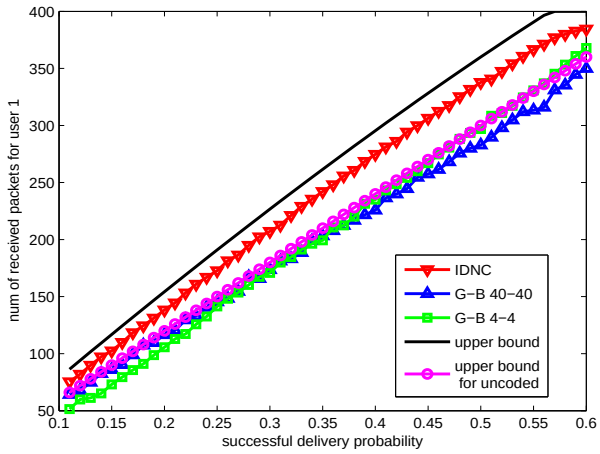

Fig. 3. Total number of received packets in session 1 when $\lambda_1 = 3$, $\lambda_2 = 3$, $N_1 = 400$, $N_2 = 400$ averaged in 100 simulations.

involving user 2. The BS can now combine the two coding opportunities and send $[X + Y]$, which serves two receivers simultaneously (and is thus more efficient than traditional uncoded retransmission). Note that in this example, the desired packet X (resp. Y) can be *immediately decoded* by d_1 (resp. d_2) upon receiving $[X + Y]$. Compared to the generation-based solutions, the IDNC schemes have zero decoding delay, and incur substantially lower encoding complexity since only binary field is used. As a result, IDNC schemes generally demonstrate much faster startup phase [24], and are more suitable for time-sensitive applications.

However, designing IDNC scheme for the setting in this work is difficult. In a single-multicast setting, the above simple IDNC scheme proposed in [6] turns out to be optimal even with deadline constraints. However, when performing coding over 2-unicast sessions, we need to take into account new issues. For example, in the under-provisioned scenarios (Figs. 1(a,c,d)) the system simply cannot sustain the rate vector $(\frac{1}{\lambda_1}, \frac{1}{\lambda_2})$. In a similar way as in the modified GB scheme in Section IV, we thus need to incorporate a new early-dropping mechanism in the IDNC scheme, the details of which would be discussed shortly after.

In addition to the challenges from the "under-provisioned case", we may also face a second challenge that arises from the heterogeneity of the channels and the deadlines, and that is orthogonal from the previous problem due to the under-provisioned scenario. More specifically, consider an over-provisioned scenario for which we can send at rate $(r_1, r_2) = (\frac{1}{\lambda_1}, \frac{1}{\lambda_2})$ that satisfy both (2) and (3). In an IDNC scheme, each packet is sent repeatedly in an uncoded fashion until it is received by at least one user. As a result, on average it takes $\frac{r_1 T}{1-(1-p_1)(1-p_2)}$ time slots to finish sending all session-1 packets uncodedly. For each time slot, with probability $p_2(1-p_1)$ such a packet will be heard only by d_2, which creates a coding opportunity involving user 1. On average, the average amount of coding opportunities of user 1 is $\frac{r_1 T \cdot p_2(1-p_1)}{1-(1-p_1)(1-p_2)}$. Note that such a coding opportunity of use 1 will later be combined with that of user 2. Note that when sending a coded packet, it takes on average $\frac{1}{p_1}$ before it can be received by d_1. Therefore, it takes $\frac{r_1 T \cdot p_2(1-p_1)}{p_1(1-(1-p_1)(1-p_2))}$ trials of sending coded packets to fully "consume the coding opportunities of user 1". Symmetrically, the average amount of time slots to fully consume the coding opportunities of user 2 is $\frac{r_2 T \cdot p_1(1-p_2)}{p_2(1-(1-p_1)(1-p_2))}$. If we have

$$\frac{r_1 T \cdot p_2(1-p_1)}{p_1(1-(1-p_1)(1-p_2))} > \frac{r_2 T \cdot p_1(1-p_2)}{p_2(1-(1-p_1)(1-p_2))}$$
$$\Leftrightarrow \frac{\lambda_1 p_1(p_1 - p_1 p_2)}{\lambda_2 p_2(p_2 - p_1 p_2)} < 1, \tag{7}$$

then from our previous arguments, it takes longer to consume all user-1 coding opportunities than to consume the coding opportunities of user 2. Those "leftover" user-1 coding opportunities (those that could not be combined with that of the user-2 coding opportunities) thus needs to be transmitted in an uncoded manner. If there is no deadline constraint, then we can simply wait until the very end (when the coding opportunities of user 2 have been used up) to decide which are the leftover user-1 coding opportunities. However, if there is deadline, when we know for sure which user-1 coding opportunities are the leftover ones, those packets may have already expired and cannot be sent anymore. The throughput thus suffers from not being able to send those leftover coding opportunities uncodedly. Note that such a challenge does not arise in the homogeneous setting of all existing IDNC work [5], [6], for which there is no left-over coding opportunity. To recover from this sub-optimality, when (7) is satisfied, *an optimal IDNC scheme should continue sending some user-1 packet in an uncoded manner even after it has been overheard by user 2.* For future reference, we say "user 1 is a leading user" if (7) is satisfied since user 1 now has more coding opportunities than that could be combined with user 2's coding opportunities. For the following, we combine the above two intuitions and design a new IDNC scheme that is capable of achieving the upper bound of deadline-constrained capacity given in Proposition 1.

To begin with, we will introduce some definitions. In our new IDNC scheme, the BS keeps two registers n_1 and n_2. One can view the purpose of n_i as to keep track of the next uncoded packet to be sent for session i. Since both n_1 and n_2 evolve over time, we sometimes use $n_i(t)$ to denote the value of n_i at the end of time t. The BS also keeps two lists of packets: L_{10} and L_{01}. List L_{01} contains all unexpired coding opportunities of user 1 (those heard by d_2 but not yet by d_1). Symmetrically, list L_{10} contains all unexpired coding opportunities of user 2. Each packet is also associated with a status, which can take one of the following four values "not-processed", "dropped", "uncoded-Tx-only" and "coding-eligible". The BS uses two arrays status1$[i]$, $i = 1, \cdots, N_1$, and status2$[i]$, $i = 1, \cdots, N_2$ to keep track of the status of the session-1 and session-2 packets, respectively. In addition, the BS keeps 4 floating-point registers, denoted by x_1, x_2, y_1, and y_2. We also assume that at the end of each time slot, both users send an ACK or NACK message back to the BS depending on whether that user has successfully received the transmitted packet in the present time slot.

In the following, we describe our IDNC scheme in details. In the time origin, the BS first initializes the following variables: $n_1 \leftarrow 1$, $n_2 \leftarrow 1$, $L_{10} \leftarrow \emptyset$, $L_{01} \leftarrow \emptyset$, status1$[i]$ $\leftarrow$not-processed, status2$[i]$ $\leftarrow$not-processed, for all i; $x_1, y_1, x_2, y_2 \leftarrow 0$. For convenience, we use γ to denote the constant value as defined in (6). The detailed steps are now described as follows.

1: **for** $t = 1$ to $\lambda_1 N_1$ **do**
2: In the beginning of time t, run the sub-routine SCHEDULE-PACKET-TRANSMISSION
3: In the end of time t, run the sub-routine UPDATE-PACKET-STATUS
4: **end for**

The two sub-routines are described separately as follows.

§ SCHEDULE-PACKET-TRANSMISSION

1: **if** $n_2 \leq N_2$ & $n_1 \leq N_1$ **then**
2: **while** status1$[n1]$ =not-processed **do**
3: $x_1 \leftarrow x_1 + \min(\gamma, 1)$
4: **if** $\lfloor x_1 \rfloor > y_1$ where $\lfloor \cdot \rfloor$ is the floor function **then**
5: $y_1 \leftarrow \lfloor x_1 \rfloor$
6: Generate a number a independently and uniformly randomly from $[0, 1]$
7: **if** $a < \frac{\lambda_1 p_1 (p_1 - p_1 p_2)}{\lambda_2 p_2 (p_2 - p_1 p_2)}$ **then**
8: status1$[n1]$ $\leftarrow$ coding-eligible
9: **else**
10: status1$[n1]$ $\leftarrow$ uncoded-Tx-only
11: **end if**
12: **else**
13: status1$[n1]$ $\leftarrow$dropped
14: $n_1 \leftarrow n_1 + 1$
15: **end if**
16: **end while**
17: Repeat the steps from Line 2 to Line 16 with the roles of users 1 and 2 swapped, i.e, we focus on user 2 now.
18: **if** both L_{10} and L_{01} are non-empty **then**
19: Choose the oldest packet X_{1,j_1^*} from L_{01} and the oldest packet X_{2,j_2^*} from L_{10}. Broadcast the sum $[X_{1,j_1^*} + X_{2,j_2^*}]$.
20: **else**
21: **if** $n_1\lambda_1 \leq n_2\lambda_2$ **then**
22: Send uncoded packet X_{1,n_1} directly.
23: **else if** $n_1\lambda_1 > n_2\lambda_2$ **then**
24: Send uncoded packet X_{2,n_2} directly.
25: **end if**
26: **end if**
27: **else**
28: Choose the oldest unexpired packets in the system (including those in $L_{01} \cup L_{10}$ and those haven't been sent) and send that packet uncodedly.
29: **end if**

§ UPDATE-PACKET-STATUS

1: **if** an uncoded packet X_{1,n_1} was sent in the current time slot **then**
2: **if** X_{1,n_1} is received by d_1 **then**
3: $n_1 \leftarrow n_1 + 1$.
4: **else if** X_{1,n_1} was received only by d_2 and status1$[n_1]$ =coding-eligible **then**
5: Add X_{1,n_1} to L_{01} and set $n_1 \leftarrow n_1 + 1$
6: **end if**
7: **else if** an uncoded packet X_{2,n_2} was sent in the current time slot **then**
8: Repeat the steps from Line 2 to Line 6 with the roles of users 1 and 2 swapped.
9: **else**
10: Suppose the coded packet being sent is $[X_{1,j_1^*} + X_{2,j_2^*}]$, the sum of X_{1,j_1^*} and X_{2,j_2^*}.
11: **if** $[X_{1,j_1^*} + X_{2,j_2^*}]$ was received by d_1 **then**
12: Remove X_{1,j_1^*} from L_{01}.
13: **end if**
14: **if** $[X_{1,j_1^*} + X_{2,j_2^*}]$ was received by d_2 **then**
15: Remove X_{2,j_2^*} from L_{10}.
16: **end if**
17: **end if**
18: Remove all expired packets from the system.

The high-level ideas of the proposed IDNC scheme is as follows. Let us first focus on the sub-routine SCHEDULE-PACKET-TRANSMISSION. Line 1 checks whether we have reached the terminal phase of the transmission, i.e., when either $n_1 > N_1$ or $n_2 > N_2$ holds, we simply choose the oldest available packet to transmit. When we are in the main loop of the transmission (the normal operations), i.e., when both $n_1 \leq N_1$ and $n_2 \leq N_2$ hold, we first assign the packet status for both X_{1,n_1} and X_{2,n_2}. More specifically, in Lines 2 to 16, we first consider the "next-to-be-transmitted" packet and will assign the corresponding packet status. To do so, we use the variables x_1 and y_1 to decide whether we would like to set the current status to "dropped". As can be easily seen in Lines 3, 4, and 13, when $\gamma \geq 1$, we never drop a packet (i.e., no packets are set to dropped). The value of γ is indeed to decide whether the system is over-provisioned ($\gamma \geq 1$) or

under-provisioned ($\gamma < 1$). As explained in Section IV, we drop a packet only when $\gamma < 1$, and Lines 3 to 5 make sure that the packet dropping ratio is equal to the pre-computed γ as in (6). If we decide to drop the packet, then we need to move on and decide the status of the next packet, see Lines 13 and 14. For those packets that are not dropped and thus will be transmitted later, we sometimes need to preemptively send those packets in an uncoded manner for the "leading user" as explained earlier in Section V. If user 1 is the leading user, then $\frac{\lambda_1 p_1(p_1-p_1p_2)}{\lambda_2 p_2(p_2-p_1p_2)} < 1$. Lines 6 to 11 ensure that some user-1 packets have their status set to uncoded-Tx-only. Note that if user 2 is the leading user, then $\frac{\lambda_1 p_1(p_1-p_1p_2)}{\lambda_2 p_2(p_2-p_1p_2)} > 1$ and Lines 6 to 11 automatically ensure that all user-1 packets have their status set to coding-eligible. Once we finish setting the packet status, we give priority to transmitting the coded packet first (Lines 18 and 19). If sending coded packets is not possible, then we evenly alternate between sending uncoded packets for users 1 and 2, by comparing the values of $n_1\lambda_1$ and $n_2\lambda_2$ (Lines 21 to 25). Namely, we choose the next uncoded packet depending on which is the closest to expire. This observation also leads to the following self-explanatory lemma.

Lemma 1. *For any time slot t, we have* $-\max(\lambda_1, \lambda_2) \leq \lambda_1 n_1(t) - \lambda_2 n_2(t) \leq \max(\lambda_1, \lambda_2)$.

Let us now focus on the sub-routine UPDATE-PACKET-STATUS. If an uncoded packet X_{1,n_1} was sent and received by d_1 (see Lines 2–3), then there is no need to retransmit this packet. We simply shift our focus to the next packet ($n_1 \leftarrow n_1 + 1$). If X_{1,n_1} is received by d_2 but not by d_1, then this packet may become a new coding opportunity. However, as mentioned earlier, if user 1 is the leading user, then sometimes we need to forgo an coding opportunity and continue sending it in an uncoded manner. This is decided by the packet status. If packet status was set to uncoded-Tx-only, then we do not put the overheard packet X_{1,n_1} in the coding list L_{01}. That is, X_{1,n_1} will not participate in any future coding operations and will still be transmitted uncodedly next time. Only when the packet status is coding-eligible (see Line 4) will the overheard X_{1,n_1} be put into the list L_{01}. Lines 11 to 18 simply perform packet update to remove the packets that have either expired or have already been decoded by the target user.

The IDNC scheme has zero decoding delay, i.e., upon the reception of any coded or uncoded packet, the user can decode one more packet for its own session. Further, the coded transmissions are mingled with the uncoded transmissions, not like the generation-based scheme. Thus the BS does not need to drop the first generation in order to let packets of the subsequent generations meet the deadline. We can clearly see from Fig. 2 and 3 that our IDNC scheme universally outperforms the generation-based schemes for either large file size and small file size. Next we would prove the asymptotic throughput optimality for our IDNC scheme.

VI. MAIN RESULT: PERFORMANCE ANALYSIS OF THE NEW IDNC SCHEME

The IDNC scheme is easier to implement than the generation based scheme in practice. However, the analysis of the IDNC scheme is rather difficult, especially under the sequential hard deadline constraints. To the best of our knowledge, our work is the first one to analyze the performance of IDNC schemes for two unicasts under sequential hard deadline constraints. The performance of the proposed new IDNC scheme is characterized as follows.

Proposition 2. *For any given system parameters p_1, p_2, λ_1, and λ_2, let β^* denote the largest β value such that $0 \leq \beta \leq 1$ and the rate vector $(r_1, r_2) = \left(\frac{\beta}{\lambda_1}, \frac{\beta}{\lambda_2}\right)$ satisfies both* (2) *and* (3)*. For any $\epsilon > 0$, there exists a sufficiently large N_1 (and $N_2 = \frac{\lambda_1 N_1}{\lambda_2}$) such that the proposed IDNC scheme achieves $\mathsf{E}\{N_1^{success}\}/N_1 \geq \frac{\beta^*}{\lambda_1} - \epsilon$ and $\mathsf{E}\{N_2^{success}\}/N_2 \geq \frac{\beta^*}{\lambda_2} - \epsilon$.*

Proposition 2 shows that our IDNC scheme achieves asymptotically the upper bound in Proposition 1 for both over-provisioned ($\beta^* = 1$) and under-provisioned ($\beta^* < 1$) scenarios. Before proving Proposition 2, we present Lemma 2, which is critical to our proof.

Lemma 2. *Consider our IDNC scheme with system parameter values λ_1, λ_2, p_1, and p_2. Then for any $\epsilon > 0$, there exists $B > 0$ such that for all fixed t_1 and t_2 satisfying $(t_2 - t_1) \geq B$, we have for $j = 1, 2$,*

$$\mathsf{E}\Big\{ n_j(t_2) - n_j(t_1) \Big| t_2 < \min(\lambda_1 n_1(t_1), \lambda_2 n_2(t_1)) \Big\} \leq \frac{(t_2 - t_1)\max(\gamma, 1)(1+\epsilon)}{\lambda_j}. \quad (8)$$

The detailed proof for Lemma 2 can be found in Appendices A and B. The high-level interpretation of this lemma is provided as follows. Consider any two fixed time instants t_1 and t_2, and assume that we are in a critically provisioned scenario: $\gamma = 1$. For $j = 1$, the term $n_1(t_2) - n_1(t_1)$ quantifies how many new session-1 packets have been "injected" to the system during the time interval $(t_1, t_2]$. Lemma 2 shows that this value cannot grow much faster than $\frac{(t_2 - t_1)}{\lambda_1}$. In other words, the growth of $n_1(t)$ in a critically-provisioned scenario is proportional to how fast the packets of session 1 expire. The sketch of the proof is as follows. Note that when conditioning on $t_2 < \min(\lambda_1 n_1(t_1), \lambda_2 n_2(t_1))$, none of these newly injected packets $X_{1,n_1(t_1)}, X_{1,n_1(t_1)+1}, \cdots, X_{1,n_1(t_2)-1}$ will expire during the interval $(t_1, t_2]$. Therefore, those packets will have similar behavior as if in a system without deadline constraints. Then, by the law of large numbers (recall that $t_2 - t_1 \geq B$ is sufficiently large), we can explicitly quantify/upper-bound the numbers of uncoded and coded transmissions in this time interval $(t_1, t_2]$, which in turn give us the inequality in (8). For the following, we would first present the proof for Proposition 2 based on Lemma 2.

Proof: For the following, we would first discuss the critically-provisioned case ($\gamma = 1$ and recall the definition of γ in (6)). We would later generalize the proof for the under-provisioned case, and the proof for the over-provisioned case[4].

[4]For a deadline constrained system, it is more interesting to quantify the performance in the under-provisioned setting because in an over-provisioned setting (when deadline is very far and each packet has plenty of time to finish transmission) even a sub-optimal scheme can easily finish transmitting all packets without violating the deadlines.

For ease of exposition, we first assume that user 1 is the leading user. Since we are considering the critically-provisioned case, we have $\frac{1}{\gamma} = \frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1+p_2-p_1p_2} = 1$. For any given $\epsilon > 0$, we use B to represent the B value specified in Lemma 2. We will describe how to choose the ϵ value in the later part of this proof. For a given $\epsilon > 0$, we define $q_j(t) \triangleq n_j(t) - \frac{\gamma t(1+2\epsilon)}{\lambda_j}$ for $j = 1, 2$. We first note that $n_j(t)$, the index of the next to-be-sent uncoded packets must satisfy $n_j(t) \geq \frac{t}{\lambda_j}$. By definition, $q_j(t)$ is thus always non-negative.

We first show that $q_1(t)$ and $q_2(t)$ cannot be very large due to Lemma 2. Consider a (t_1, t_2) pair satisfying $B_0 \triangleq t_2 - t_1 > B$. Note that by the definition of $q_1(t)$, $q_2(t)$, and by Lemma 1, we have $q_2(t) \geq \frac{n_1(t)\lambda_1 - \lambda_1}{\lambda_2} - \frac{\gamma t(1+2\epsilon)}{\lambda_2} = \frac{\lambda_1}{\lambda_2} q_1(t) - \frac{\lambda_1}{\lambda_2}$. This observation thus implies that if $q_1(t_1) > \frac{B_0}{\lambda_1} + 1$, then $q_2(t_1) > \frac{B_0}{\lambda_2}$. One can also check that if the following three conditions $q_1(t_1) > \frac{B_0}{\lambda_1} + 1$ and $q_2(t_1) > \frac{B_0}{\lambda_2}$, $t_2 = t_1 + B_0$ hold simultaneously, then $t_2 < \min(\lambda_1 n_1(t_1), \lambda_2 n_2(t_1))$ in Lemma 2.

Note that by the definition of $q_1(t)$, we can see that the condition $q_1(t_1) > \frac{B_0}{\lambda_1} + \max\left(1, \frac{\lambda_2}{\lambda_1}\right)$ implies that $\lambda_1 n_1(t_1) - \max(\lambda_1, \lambda_2) > t_2$. By Lemma 1, this further implies that $t_2 \leq \max(\lambda_1 n_1(t_1), \lambda_2 n_2(t_1))$. We then have

$$\begin{aligned}
&\mathsf{E}\Big\{q_1(t_1 + B_0) - q_1(t_1)\Big|q_1(t_1) > \frac{B_0}{\lambda_1} + \max(1, \frac{\lambda_2}{\lambda_1})\Big\} \\
&= \mathsf{E}\Big\{n_1(t_1 + B_0) - n_1(t_1)\Big|q_1(t_1) > \frac{B_0}{\lambda_1} + \max(1, \frac{\lambda_2}{\lambda_1})\Big\} \\
&\quad - \frac{B_0(1+2\epsilon)}{\lambda_1} \qquad (9) \\
&\leq \frac{B_0\gamma(1+\epsilon)}{\lambda_1} - \frac{B_0\gamma(1+2\epsilon)}{\lambda_1} < 0, \qquad (10)
\end{aligned}$$

where (9) follows from the definition of $q_1(t)$, the first inequality of (10) follows from Lemma 2. Eq. (10) shows that $q_1(t)$ has a negative drift. As a result, for any $\epsilon_1, \epsilon_2 > 0$, there exists a $t_0 > 0$ such that

$$\forall t > t_0, \quad \mathsf{P}\left(q_1(t) < \epsilon_1 t\right) > 1 - \epsilon_2. \qquad (11)$$

Now that we have shown that $q_1(t)$ and $q_2(t)$ cannot be very large, we next show that $n_1(t)$ and $n_2(t)$ cannot be made much larger than $\frac{t}{\lambda_1}$ and $\frac{t}{\lambda_2}$, respectively, either. Specifically, the following inequality holds for any $t > t_0$,

$$\begin{aligned}
\mathsf{E}\{n_1(t)\} &= \mathsf{E}\left\{\frac{\gamma t(1+2\epsilon)}{\lambda_1} + q_1(t)\right\} \\
&= \mathsf{E}\left\{\frac{\gamma t(1+2\epsilon)}{\lambda_1} + q_1(t)\Big|q_1(t) < \epsilon_1 t\right\}\mathsf{P}(q_1(t) < \epsilon_1 t) \\
&\quad + \mathsf{E}\{n_1(t)|q_1(t) \geq \epsilon_1 t\}\mathsf{P}(q_1(t) \geq \epsilon_1 t) \\
&\leq \Big(\frac{\gamma t(1+2\epsilon)}{\lambda_1} + \epsilon_1 t\Big) + t\epsilon_2, \qquad (12)
\end{aligned}$$

where (12) is because $n_1(t)$ is always upper bounded by t regardless whether $q_1(t) \geq \epsilon_1 t$ or not. Note that we can choose arbitrarily small ϵ, ϵ_1, and ϵ_2 and (12) still holds for sufficiently large t. As a result, (12) shows that the expectation $\mathsf{E}\{n_1(t)\}$ is upper bounded by $\frac{\gamma t}{\lambda_1} + o(t)$. Similarly, we can prove $\mathsf{E}\{n_2(t)\} \leq \frac{\gamma t}{\lambda_2} + o(t)$ ($\gamma = 1$ for critically-provisioned case).

We next use these inequalities to bound the number of successful transmissions to user 1 and 2. For the following, we temporarily assume that the file sizes are infinity by adding dummy packets to both sessions, which are labeled as $X_{j,N_j+1}, X_{j,N_j+2}, \cdots$ for $j = 1, 2$. In this way, we can continue executing Lines 2 to 26 of SCHEDULE-PACKET-TRANSMISSION without worrying about the degenerated cases when executing Line 28. We then define $T_j(t)$ as the number of time slots when the BS transmits an uncoded packet for session j up to time t (those time slots when Lines 22 or 24 of SCHEDULE-PACKET-TRANSMISSION are executed). Since user 2 is not the leading user, the BS transmits every session-2 packet uncodedly until it has been received by at least one user. We thus have

$$\mathsf{E}\{T_2(t)\} \leq \mathsf{E}\{n_2(t)\}\frac{1}{p_1 + p_2 - p_1p_2}, \qquad (13)$$

where the inequality is because some uncoded packets are expired before they can be received by any user, and hence the expected transmission time for each packet is no larger than the case when there is no expiration. Next, we consider $T_1(t)$. Note that for session 1, some packets would be transmitted repetitively until user 1 receives it even after it has been received by user 2. $T_1(t)$ is thus comprised of two types of transmissions: The first type counts the number of time slots in which the BS transmits an uncoded packet of session 1 that has not been heard by any user. The second type counts the number of time slots in which the BS transmits a session-1 packet uncodedly even though that packet has been heard by user 2 already (due to its status being set to uncoded-Tx-only and in which case the BS continues to transmit this packet until user 1 receives it). The first part can be upper bounded by $\mathsf{E}\{n_1(t)\}\frac{1}{p_1+p_2-p_1p_2}$ in the same way as in (12). We use $\mathsf{UCO}(t)$ to denote the total number of the second type of uncoded-Tx-only transmission during the interval $[1, t]$. We then have

$$\begin{aligned}
\mathsf{E}\Big\{\mathsf{UCO}(t)\Big\} &\leq \mathsf{E}\{n_1(t)\}\Big(1 - \frac{\lambda_1 p_1(p_1 - p_1p_2)}{\lambda_2 p_2(p_2 - p_1p_2)}\Big) \\
&\quad \times \left(\frac{p_2(1-p_1)}{1-(1-p_1)(1-p_2)}\right)\frac{1}{p_1}. \qquad (14)
\end{aligned}$$

The explanation of (14) is as follows. Out of all $n_1(t)$ session-1 packets that have been transmitted during time interval $[1, t]$, a fraction of $(1 - \frac{\lambda_1 p_1(p_1-p_1p_2)}{\lambda_2 p_2(p_2-p_1p_2)})$ has their status set to uncoded-Tx-only. Out of those with status set to uncoded-Tx-only, a fraction of $\left(\frac{p_2(1-p_1)}{1-(1-p_1)(1-p_2)}\right)$ will be heard by d_2 first (strictly before it is heard by d_1). For those that have been heard by d_2 first, it takes, on average, additional $\frac{1}{p_1}$ time slots of transmission before it can be heard by the intended user d_1. The inequality is again to take into account that some packets may expire even before finishing its corresponding transmission. Combining the first and second part, we obtain

$$\mathsf{E}\{T_1(t)\} \leq \mathsf{E}\{n_1(t)\}\frac{1}{p_1}\left(1 - \frac{\lambda_1 p_1(p_1 - p_1p_2)}{\lambda_2 p_2(p_1 + p_2 - p_1p_2)}\right). \qquad (15)$$

Note that when we transmit an uncoded packet for session 1, the expected "reward" is p_1 since only user 1 can benefit from this transmission. When we transmit a coded packet, the expected reward for user 1 is p_1 and the expected reward for user 2 is p_2 since both destinations can benefit from the coded transmission. Note that by definition the total number of coded transmission in the $[1,t]$ interval is $t-T_1(t)-T_2(t)$. We can now lower bound the expected total rewards for user 1:

$$\begin{aligned}&\mathsf{E}\{N_1^{\text{success}}\} = p_1\mathsf{E}\{T_1(t)\} + p_1\mathsf{E}\{t-T_1(t)-T_2(t)\}\\&= p_1 t - p_1\mathsf{E}\{T_2(t)\}\\&\geq p_1 t - p_1\frac{\gamma t}{\lambda_2}\frac{1}{p_1+p_2-p_1p_2} - o(t)\\&= \frac{\gamma t}{\lambda_1} - o(t),\end{aligned} \tag{16}$$

where the inequality follows from $\mathsf{E}\{n_2(t)\} \leq \frac{\gamma t}{\lambda_2} + o(t)$ and (13), and (16) follows from plugging in the definition of γ and arithmetic simplification (here $\gamma = 1$).

Consider the asymptotic regime with sufficiently large N_1 and N_2. We choose $t = t^* \triangleq \frac{\lambda_1 N_1}{\gamma(1+3\epsilon)}$, and we have $\mathsf{E}\{N_1^{\text{success}}\} = \frac{N_1}{1+3\epsilon} - o(t)$. Recall that the above expected reward $\mathsf{E}\{N_1^{\text{success}}\}$ is the number of session-1 packets that are successfully decoded by user 1 by the end of time t^* and *that also counts the dummy packets X_{j,N_j+1}, X_{j,N_j+2},$\cdots$ added after X_{j,N_j}* (so that we can avoid executing Line 28). To count only the real packets, we notice that, by (11), for sufficiently large N_1, with high probability $1-\epsilon_2$ we have

$$\begin{aligned}&n_1(t^*) - \frac{\gamma t^*(1+2\epsilon)}{\lambda_1} = q_1(t*) < \epsilon_1 t^*\\&\Leftrightarrow n_1(t^*) < N_1\frac{1+2\epsilon}{1+3\epsilon} + \epsilon_1 t^*.\end{aligned} \tag{17}$$

By choosing sufficiently small ϵ_1, the above analysis shows that $n_1(t^*) < N_1$ with probability $\geq 1-\epsilon_2$. Symmetrically, $P(n_2(t^*) < N_2) \geq 1-\epsilon_2$. Jointly, $P(n_1(t^*) < N_1, n_2(t^*) < N_2) \geq 1-2\epsilon_2$. It means that at time t^* with high probability $1-2\epsilon_2$, both indices $n_1(t^*)$ and $n_2(t^*)$ are still less than N_1 and N_2, respectively. Therefore, no dummy packets have been injected in the system yet. As a result, even when we run the algorithm without any dummy packets, the expected success $\mathsf{E}\{N_1^{\text{success}}(\text{without dummy packets})\}$ at time t^* must be no smaller than $\frac{1}{1-2\epsilon_2}(\mathsf{E}\{N_1^{\text{success}}(\text{with dummy packets})\} - 2\epsilon_2 N_1)$. Choosing sufficiently small ϵ and ϵ_2, we have thus $\frac{\mathsf{E}\{N_1^{\text{success}}(\text{without dummy packets})\}}{N_1}$ approaches 1 when both N_1 and N_2 are sufficiently large.

Similarly, we can prove $\frac{\mathsf{E}\{N_2^{\text{success}}(\text{without dummy packets})\}}{N_2}$ approaches 1 when N_1 and N_2 are sufficiently large. The expected total rewards for user 2 can be lower bounded by

$$\begin{aligned}&\mathsf{E}\{N_2^{\text{success}}\} = p_2\mathsf{E}\{T_2(t)\} + p_2\mathsf{E}\{t-T_1(t)-T_2(t)\}\\&= p_2 t - p_2\mathsf{E}\{T_1(t)\}\\&\geq p_2 t - p_2\frac{\gamma t}{\lambda_1}\left(\frac{1}{p_1} - \frac{N_2(p_1-p_1p_2)\frac{1}{p_2}}{(p_1+p_2-p_1p_2)N_1}\right) - o(t)\\&= \frac{\gamma t}{\lambda_2} - o(t).\end{aligned} \tag{18}$$

When $t = \lambda_2 N_2/\gamma$, we have $\mathsf{E}\{N_2^{\text{success}}\} = N_2 - o(t)$. Hence, the achievable rate $\frac{N_2^{\text{success}}}{\lambda_2 N_2}$ also approaches $\frac{1}{\lambda_2}$ for sufficiently large N_2. By the similar arguments for the "dummy packets" analysis of user 1, we can also prove the throughput optimality of user 2.

The case when user 2 is the leading user can be proved similarly.

We have shown the optimality proof for the critically-provisioned case. Next, we are going to show the maximum throughput that can be achieved by our scheme for the under-provisioned case $\gamma < 1$. For ease of exposition, we assume that user 1 is the leading user. Define $q_1(t) \triangleq n_1(t) - \frac{t(1+2\epsilon)}{\lambda_1}$ and $q_2(t) \triangleq n_2(t) - \frac{t(1+2\epsilon)}{\lambda_2}$. Then in the same way as in the critical-provisioned case, we can prove the negative drift of $q_1(t)$ and $q_2(t)$ and consequently prove the existence of t_0 such that for any $t > t_0$, $\mathsf{E}\{n_1(t)\} \leq \frac{t}{\lambda_1} + o(t)$ and $\mathsf{E}\{n_2(t)\} \leq \frac{t}{\lambda_2} + o(t)$.

Compared to the critically-provisioned case, the main difference is that for the under-provisioned case, a new packet-dropping mechanism is used in Line 2 to Line 16 of SCHEDULE-PACKET-TRANSMISSION. Therefore, we need to carefully take that into account in our analysis. Use the same definition of $T_1(t)$ and $T_2(t)$ as in the previous proof, we can upper bound $\mathsf{E}\{T_2(t)\}$ as follows. By our dropping mechanism for the under provisioned case, we can upper bound $\mathsf{E}\{T_2(t)\}$ easily. Since user 1 is the leading user, $T_1(t)$ is still comprised of two parts: one part is when the BS transmits uncoded packets of session 1, the other part is when a session 1 packet has been received by user 2 first, the BS continues to transmit this packet until user 1 receives it. We can upper bound the first part and second part separately, and then upper bound $\mathsf{E}\{T_1(t)\}$. By the same argument as in the previous proof, expected total rewards for users 1 and 2 are lower bounded by

$$\mathsf{E}\{N_1^{\text{success}}\} = \frac{t}{\lambda_1}\Big(\frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1+p_2-p_1p_2}}\Big) + o(t), \tag{19}$$

and

$$\mathsf{E}\{N_2^{\text{success}}\} = \frac{t}{\lambda_2}\Big(\frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1+p_2-p_1p_2}}\Big) + o(t). \tag{20}$$

The remaining step is again to show that at time $t^* \triangleq \frac{\lambda_1 N_1}{1+3\epsilon}$, with close to one probability both $n_1(t^*) < N_1$ and $n_2(t^*) < N_2$. Therefore, the two equations guarantees that $\frac{\mathsf{E}\{N_j^{\text{success}}\}}{N_j}$ approaches $\frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1+p_2-p_1p_2}}$ for sufficiently large N_1 and N_2.

Next we show proof for the over-provisioned case. For ease of exposition, we first assume that user 1 is the leading user. Since we are considering the over-provisioned case, we have $\frac{1}{\gamma} = \frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1+p_2-p_1p_2} > 1$. For any given $\epsilon > 0$, we use B to represent the B value specified in Lemma 2. We will describe how to choose the ϵ value in the later part of this proof. For a given $\epsilon > 0$, we define $q_j(t) \triangleq n_j(t) - \frac{\gamma t(1+2\epsilon)}{\lambda_j}$ for $j = 1, 2$.

We first show that $q_1(t)$ and $q_2(t)$ cannot be very large due to Lemma 2. Consider a (t_1, t_2) pair satisfying $B_0 \triangleq t_2 - t_1 > B$. Note that by the definition of $q_1(t)$, $q_2(t)$, and by

Lemma 1, we have $q_2(t) \geq \frac{n_1(t)\lambda_1-\lambda_1}{\lambda_2} - \frac{\gamma t(1+2\epsilon)}{\lambda_2} = \frac{\lambda_1}{\lambda_2}q_1(t) - \frac{\lambda_1}{\lambda_2}$. This observation thus implies that if $q_1(t_1) > \frac{B_0}{\lambda_1}+1$, then $q_2(t_1) > \frac{B_0}{\lambda_2}$. One can also check that if the following three conditions $q_1(t_1) > \frac{B_0}{\lambda_1}+1$ and $q_2(t_1) > \frac{B_0}{\lambda_2}$, $t_2 = t_1 + B_0$ hold simultaneously, then $t_2 < \min(\lambda_1 n_1(t_1), \lambda_2 n_2(t_1))$ in Lemma 2.

Note that by the definition of $q_1(t)$, we can see that the condition $q_1(t_1) > \frac{B_0}{\lambda_1} + \max\left(1, \frac{\lambda_2}{\lambda_1}\right)$ implies that $\lambda_1 n_1(t_1) - \max(\lambda_1, \lambda_2) > t_2$. By Lemma 1, this further implies that $t_2 \leq \max(\lambda_1 n_1(t_1), \lambda_2 n_2(t_1))$. Then, by the same arguments for the critical-provisioned case, we have

$$\begin{aligned}
&\mathsf{E}\Big\{q_1(t_1+B_0) - q_1(t_1)\Big|q_1(t_1) > \frac{B_0}{\lambda_1} + \max(1, \frac{\lambda_2}{\lambda_1})\Big\} \\
&= \mathsf{E}\Big\{n_1(t_1+B_0) - n_1(t_1)\Big|q_1(t_1) > \frac{B_0}{\lambda_1} + \max(1, \frac{\lambda_2}{\lambda_1})\Big\} \\
&\quad - \frac{B_0(1+2\epsilon)}{\lambda_1} \qquad (21) \\
&\leq \frac{B_0\gamma(1+\epsilon)}{\lambda_1} - \frac{B_0\gamma(1+2\epsilon)}{\lambda_1} < 0. \qquad (22)
\end{aligned}$$

We have shown that $q_1(t)$ has a negative drift. As a result, for any $\epsilon_1, \epsilon_2 > 0$, there exists a $t_0 > 0$ such that

$$\forall t > t_0, \quad \mathsf{P}\left(q_1(t) < \epsilon_1 t\right) > 1 - \epsilon_2. \qquad (23)$$

Now that we have shown that $q_1(t)$ and $q_2(t)$ cannot be very large, we next show that $n_1(t)$ and $n_2(t)$ cannot be made much larger than $\frac{\gamma t}{\lambda_1}$ and $\frac{\gamma t}{\lambda_2}$, respectively, either. Specifically, the following inequality holds for any $t > t_0$,

$$\begin{aligned}
&\mathsf{E}\{n_1(t)\} = \mathsf{E}\left\{\frac{\gamma t(1+2\epsilon)}{\lambda_1} + q_1(t)\right\} \\
&= \mathsf{E}\left\{\frac{\gamma t(1+2\epsilon)}{\lambda_1} + q_1(t)\Big|q_1(t) < \epsilon_1 t\right\}\mathsf{P}(q_1(t) < \epsilon_1 t) \\
&\quad + \mathsf{E}\{n_1(t)|q_1(t) \geq \epsilon_1 t\}\mathsf{P}(q_1(t) \geq \epsilon_1 t) \\
&\leq \Big(\frac{\gamma t(1+2\epsilon)}{\lambda_1} + \epsilon_1 t\Big) + t\epsilon_2. \qquad (24)
\end{aligned}$$

By the same argument for the critical-provisioned case, (12) shows that the expectation $\mathsf{E}\{n_1(t)\}$ is upper bounded by $\frac{\gamma t}{\lambda_1} + o(t)$, and $\mathsf{E}\{n_2(t)\} \leq \frac{\gamma t}{\lambda_2} + o(t)$.

We next use these inequalities to bound the number of successful transmissions to user 1 and 2. For the following, we temporarily assume that the file sizes are infinity by adding dummy packets to both sessions, which are labeled as $X_{j,N_j+1}, X_{j,N_j+2}, \cdots$ for $j = 1, 2$. In this way, we can continue executing Lines 2 to 26 of SCHEDULE-PACKET-TRANSMISSION without worrying about the degenerated cases when executing Line 28. We then define $T_j(t)$ as the number of time slots when the BS transmits an uncoded packet for session j up to time t (those time slots when Lines 22 or 24 of SCHEDULE-PACKET-TRANSMISSION are executed). Since user 2 is not the leading user, the BS transmits every session-2 packet uncodedly until it has been received by at least one user. We thus have

$$\mathsf{E}\{T_2(t)\} \leq \mathsf{E}\{n_2(t)\}\frac{1}{p_1+p_2-p_1p_2}, \qquad (25)$$

where the inequality is because some uncoded packets are expired before they can be received by any user, and hence the expected transmission time for each packet is no larger than the case when there is no expiration. Next, we consider $T_1(t)$. Note that for session 1, some packets would be transmitted repetitively until user 1 receives it even after it has been received by user 2. $T_1(t)$ is thus comprised of two types of transmissions: The first type counts the number of time slots in which the BS transmits an uncoded packet of session 1 that has not been heard by any user. The second type counts the number of time slots in which the BS transmits a session-1 packet uncodedly even though that packet has been heard by user 2 already (due to its status being set to uncoded-Tx-only and in which case the BS continues to transmit this packet until user 1 receives it). The first part can be upper bounded by $\mathsf{E}\{n_1(t)\}\frac{1}{p_1+p_2-p_1p_2}$. We still use $\mathsf{UCO}(t)$ to denote the total number of the second type of uncoded-Tx-only transmission during the interval $[1, t]$. We then have

$$\begin{aligned}
\mathsf{E}\Big\{\mathsf{UCO}(t)\Big\} &\leq \mathsf{E}\{n_1(t)\}\Big(1 - \frac{\lambda_1 p_1(p_1 - p_1p_2)}{\lambda_2 p_2(p_2 - p_1p_2)}\Big) \\
&\quad \times \left(\frac{p_2(1-p_1)}{1-(1-p_1)(1-p_2)}\right)\frac{1}{p_1}. \qquad (26)
\end{aligned}$$

Combining the first and second part, we obtain

$$\mathsf{E}\{T_1(t)\} \leq \mathsf{E}\{n_1(t)\}\frac{1}{p_1}\left(1 - \frac{\lambda_1 p_1(p_1 - p_1p_2)}{\lambda_2 p_2(p_1+p_2-p_1p_2)}\right). \qquad (27)$$

Note that when we transmit an uncoded packet for session 1, the expected "reward" is p_1 since only user 1 can benefit from this transmission. When we transmit a coded packet, the expected reward for user 1 is p_1 and the expected reward for user 2 is p_2 since both destinations can benefit from the coded transmission. Note that by definition the total number of coded transmission in the $[1, t]$ interval is $t - T_1(t) - T_2(t)$. We can now lower bound the expected total rewards for user 1:

$$\begin{aligned}
&\mathsf{E}\{N_1^{\text{success}}\} = p_1\mathsf{E}\{T_1(t)\} + p_1\mathsf{E}\{t - T_1(t) - T_2(t)\} \\
&= p_1 t - p_1\mathsf{E}\{T_2(t)\} \\
&\geq p_1 t - p_1\frac{\gamma t}{\lambda_2}\frac{1}{p_1+p_2-p_1p_2} - o(t) \\
&= \frac{\gamma t}{\lambda_1} - o(t) \qquad (28)
\end{aligned}$$

Similarly we can show that

$$\mathsf{E}\{N_2^{\text{success}}\} \geq \frac{\gamma t}{\lambda_2} - o(t) \qquad (29)$$

The case when user 2 is the leading user can be proved similarly.

The proof for Proposition 2 is thus complete. ■

VII. SIMULATION

Our previous analyses focus on the asymptotic regime with large file sizes $N_1 \to \infty$ and $N_2 \to \infty$. In this section, we use simulation to verify the performance of our IDNC scheme for finite N_1 and N_2.

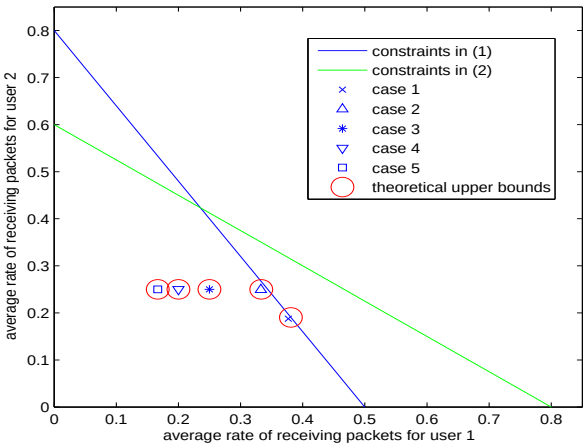

Fig. 4. Average rate of receiving packets for user 1 and user 2 when N_1 and N_2 are large.

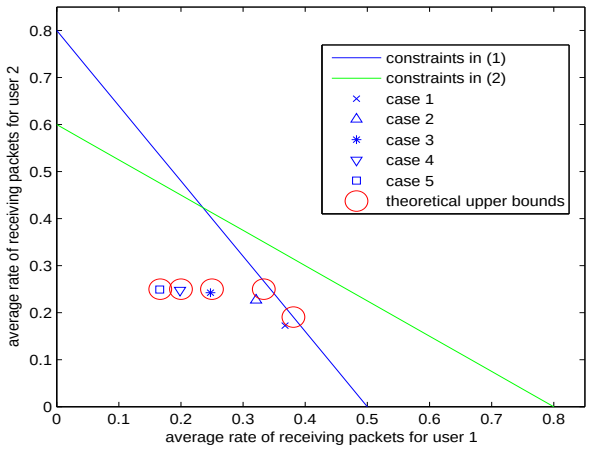

Fig. 5. Average rate of receiving packets for user 1 and user 2 when N_1 and N_2 are small.

A. Performance for Large N_1 and N_2

We first assume that the successful delivery probabilities for user 1 and user 2 are $p_1 = 0.5$ and $p_2 = 0.6$, respectively. Then we consider the following 5 cases with (λ_1, λ_2) being (2,4), (3,4), (4,4), (5,4), and (6,4), respectively (we name them as Cases 1 to 5, respectively). For all cases we use $N_1 = 40000$. Recall that we require $\lambda_1 N_1 = \lambda_2 N_2$. We thus set N_2 to be 20000, 30000, 40000, 50000, and 60000 in the 5 cases.

We first show the capacity region without deadline constraints in Fig. 4, i.e., according to (2) and (3), as shown by the area beneath the two solid lines. We then use different markers to denote the normalized throughput $(\frac{N_1^{\text{success}}}{\lambda_1 N_1}, \frac{N_2^{\text{success}}}{\lambda_2 N_2})$ from simulation for the 5 cases. The circles indicate the corresponding theoretical upper bound of both sessions, which are given by $\left(\frac{\beta^*}{\lambda_1}, \frac{\beta^*}{\lambda_2}\right)$ in Proposition 1.

More specifically, Cases 1 and 2 are the under-provisioned scenarios for which $\beta^* < 1$ and the throughput is limited by the two lines rather than by the maximum rate $(\frac{1}{\lambda_1}, \frac{1}{\lambda_2})$. Cases 3 to 5 are the over-provisioned scenarios for which the throughput is decided by the maximum rate $(\frac{1}{\lambda_1}, \frac{1}{\lambda_2})$. We observe that in all cases, the achievable throughput coincides to the theoretic upper bound, as predicted by Proposition 2.

B. Performance for Small N_1 and N_2

We are also interested in the performance of the IDNC scheme in the finite regime (when N_1 and N_2 are small). In Fig. 5 we plot the normalized throughput for both users when N_1 and N_2 are small. We use the same parameters as in Section VII-A except with smaller file sizes (N_1, N_2) being (400, 200), (400, 300), (400, 400), (400, 500), and (400, 600). We can observe that, although the numbers of packets for both session 1 and session 2 are small, the achievable throughput are still very close to the theoretical upper bound.

C. Comparison of the generation-based scheme with the IDNC scheme

In Figs. 2 and 3, we compare the performance between the generation-based scheme and the IDNC scheme. We can easily see that in Fig. 2 the performance of the IDNC scheme approaches the outer bound for the entire range of the p values. It is still true even for small file size in Fig. 3. The IDNC scheme dynamically arranges the operations of coded transmission, uncoded transmission, and drops packets in an "online" fashion, while the generation-based scheme stubbornly stick to the pre-fixed order for these operations. Moreover, the IDNC scheme takes less complexity in the encoding process, and consumes relatively smaller buffer size for storing the coding opportunities.

TABLE I
COMPARISON FOR IDNC SCHEMES WITH & WITHOUT KNOWN CHANNEL

	user-1	user-2	user-1 with estimation	user-2 with estimation
case 1	0.9431	0.8714	0.9436	0.8763
case 2	0.9551	0.9280	0.9523	0.9322
case 3	0.9844	0.9799	0.9830	0.9807
case 4	0.9905	0.9910	0.9908	0.9896
case 5	0.9952	0.9944	0.9951	0.9940

D. Extensions to The Settings of Unknown Channel

Although our proof of asymptotic optimality assumes that the BS knows the channel parameters p_1 and p_2, we believe that IDNC schemes can also achieve good performance without channel information. Specifically, since the users would send an ACK to the BS at the end of each time slot, the BS can use this feature to "estimate" the channel parameters $\hat{p}_1$ and $\hat{p}_2$ and plug them into the IDNC subroutines as a substitute for the actual p_1 and p_2. For the following, we use simulation to study the performance of the "adaptive" IDNC scheme that estimates the channel parameters on the fly. We consider the same 5 cases as in Sections VII-A and VII-B. For the cases of large N_1 and N_2 as in Section VII-A, since by the law of large numbers, the estimate $\hat{p}_j \to p_j$ for all $j = 1, 2$, the normalized throughput of the adaptive IDNC scheme is always within 1% of the performance when the values of p_1 and p_2 are known to the BS. For small N_1 and N_2 as in Section VII-B, we summarize our finding in Table I. We find that, even for small file size, the performance with channel estimation is

very close to the performance with known channel parameters. Based on the above observation, our IDNC scheme is robust and approaches the optimal throughput even when the channel parameters are unknown. Finally even if the feedback from users is not perfect, we can design a similar mechanism like in [6], to solve the problem of delayed and lossy feedback.

VIII. Conclusion and Discussion

In this work, we have studied inter-session network coding for sending two unicast sessions over an unreliable wireless channel. We have considered two unicast sessions under heterogeneous channel conditions and heterogeneous deadline constraints. We developed both a generation-based scheme and an immediately-decodable network coding (IDNC) scheme for controlling packet transmissions for the unicast sessions in order to maximize the normalized throughput subject to hard deadline constraints. The newly designed IDNC scheme is proven to be asymptotically optimal (when the file size is large), so is the generation-based scheme. Moreover, the IDNC scheme also has significantly less complexity and buffer requirements, and achieves close-to-optimal throughput even for small file sizes, an attribute not found in the generation-based solutions.

Acknowledgment

This work has been partially supported by the NSF grants CNS-0721484, CNS-0721477, CNS-0643145, CCF-0845968, CNS-0905331, and a grant from Purdue Research Foundation.

References

[1] 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.
[2] S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.
[3] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.
[4] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. of Allerton Conference*, 2003.
[5] 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.
[6] ——, "On the capacity of immediately-decodable coding schemes for wireless stored-video broadcast with hard deadline constraints," *IEEE Journal on Selected Areas in Communications, Issue on Trading Rate for Delay at the Application and Transport Layers*, vol. 29, no. 5, pp. 1094–1105, May 2011.
[7] 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.
[8] A. Ramakrishnan, A. Das, H. Maleki, A. Markopoulou, S. Jafar, and S. Vishwanath, "Network coding for three unicast sessions: Interference alignment approaches," in *Proc. of Allerton Conference*, 2010.
[9] A. Eryilmaz and D. Lun, "Control for inter-session network coding," in *NetCod*, 2007.
[10] D. L. D. Traskov, N. Ratnakar, R. Koetter, and M. Médard, "Network coding for multiple unicasts: An approach based on linear optimization," in *Proc. of ISIT*, 2006.
[11] C.-C. Wang, N. Shroff, and A. Khreishah, "Cross-layer optimizations for intersession network coding on practical 2-hop relay networks," in *Proc. of Asilomar Conference on Signals, Systems and Computers*, 2009.
[12] C.-C. Wang, "On the capacity of wireless 1-hop intersession network coding - a broadcast packet erasure channel approach," *IEEE Trans. Information Theory*, vol. 58, no. 2, pp. 957–988, Feb 2012.
[13] H. Seferoglu and A. Markopoulou, "I^{2}nc: Intra- and inter-session network coding for unicast flows in wireless networks," in *Proc. of INFOCOM*, 2011.
[14] L. Georgiadis and L. Tassiulas, "Broadcast erasure channel with feedback – capacity and algorithms," in *NetCod*, 2009.
[15] A. Eryilmaz, A. Ozdaglar, and M. Médard, "On delay performance gains from network coding," in *Proc. of CISS*, 2006.
[16] J.-K. Sundararajan, D. Shah, and M. Médard, "ARQ for network coding," in *Proc. of ISIT*, 2008.
[17] J.-K. Sundararajan, P. Sadeghi, and M. Médard, "A feedback-based adaptive broadcast coding scheme for reducing in-order delivery delay," in *NetCod*, 2009.
[18] W. Yeow, A. Hoang, and C. Tham, "Minimizing delay for multicast-streaming in wireless networks with network coding," in *Proc. of INFOCOM*, 2009.
[19] E. Drinea, C. Fragouli, and L. Keller, "Delay with network coding and feedback," in *Proc. of ISIT*, 2009.
[20] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," in *Proc. of ACM MobiCom*, 2007.
[21] D. Nguyen, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," in *NetCod*, 2007.
[22] D. Koutsonikolas, C.-C. Wang, Y. Hu, and N. Shroff, "Fec-based ap downlink transmission schemes for multiple flows: Combining the reliability and throughput enhancement of intra- and inter-flow coding," *Elsevier Performance Evaluation*, vol. 68, no. 11, pp. 1118–1135, Nov 2011.
[23] X. Li, C.-C. Wang, and X. Lin, "Optimal immediately-decodable inter-session network coding (idnc) schemes for two unicast sessions with hard deadline constraints," in *Proc. of Allerton Conference, invited paper*, 2011.
[24] J. Barros, R. Costa, D. Munaretto, and J. Widmer, "Effective delay control in online network coding," in *Proc. of INFOCOM*, 2009.

Appendix A
Proof of Lemma 2 for The Over Provisioned Case

Proof: We first present a detailed proof of Lemma 2 for the over-provisioned case, that is, $\gamma \geq 1$. By definition (6), we thus have $\frac{1}{\gamma} = \frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1+p_2-p_1p_2} \leq 1$. Without loss of generality, we assume that user 1 is the leading user, that is, $\frac{\lambda_1 p_1(p_1-p_1p_2)}{\lambda_2 p_2(p_2-p_1p_2)} < 1$. So $\frac{1}{\gamma} = \frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1+p_2-p_1p_2}$. The following discussion is conditioned on the event that in the end of time t_1, we have $\mathcal{A}_{t_1} \stackrel{\Delta}{=} \{t_2 < \lambda_1 n_1(t_1), t_2 < \lambda_2 n_2(t_1)\}$. Define

$$\Delta n_1 = \left\lfloor \frac{(t_2 - t_1)}{\left(\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1+p_2-p_1p_2}\right)\lambda_1} \right\rfloor + 1, \qquad (30)$$

$$\Delta n_2 = \left\lfloor \frac{(t_2 - t_1)}{\left(\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1+p_2-p_1p_2}\right)\lambda_2} \right\rfloor + 1. \qquad (31)$$

Note that by our definition, $\Delta n_1 \lambda_1 \approx \Delta n_2 \lambda_2$.

From the beginning of time $t_1 + 1$, let us temporarily suspend the "expiration mechanism" and use our proposed scheme to transmit packets while allowing the supposedly-expired packets to remain in the system. We first examine how long it takes before the register $n_1(t)$ evolves from its current value $n_1(t_1)$ to a different value $n_1(t_1) + \Delta n_1$, and the register $n_2(t)$ evolves from its current value $n_2(t_1)$ to a different value $n_2(t_1) + \Delta n_2$. More specifically, we use t_3 to denote the (random) time slot that is the first time slot $t \geq t_1$ such that both $n_1(t)$ is at least $n_1(t_1) + \Delta n_1$ and $n_2(t)$ is at least $n_2(t_1) + \Delta n_2$. The following proof can be divided into three corollaries. Under the assumption that the expiration mechanism is suspended from t_1 and onward, we first prove that the random variable t_3 is no less than the given constant t_2 with high probability. That is,

Corollary 1. *Without considering hard deadline constraints, for any $\epsilon > 0$, $\delta > 0$, if $t_2 - t_1$ is sufficiently large, then*

$$\mathsf{P}\left((t_3 - t_1) > (t_2 - t_1)(1-\epsilon) \middle| \mathcal{A}_{t_1}\right) > 1 - \delta. \tag{32}$$

Based on Corollary 1, we will then show that the "growth"of $n_j(t)$ from time $t_1 + 1$ to t_2 is upper bounded by $\frac{(t_2-t_1)\gamma}{\lambda_j}$:

Corollary 2. *Without considering hard deadline constraints, for any $\epsilon > 0$, there exists a sufficiently large B such that if $t_2 - t_1 > B$, then*

$$\mathsf{E}\Big\{n_j(t_2) - n_j(t_1) | \mathcal{A}_{t_1}\Big\} \leq \frac{(t_2 - t_1)\gamma(1+\epsilon)}{\lambda_j}. \tag{33}$$

Finally, we will take into account the hard deadline constraints and show that even with the hard deadline constraints, we still have

Corollary 3. *After considering hard deadline constraints, for any $\epsilon > 0$, there exists a sufficiently large B such that if $t_2 - t_1 > B$, then*

$$\mathsf{E}\Big\{n_j(t_2) - n_j(t_1) | \mathcal{A}_{t_1}\Big\} \leq \frac{(t_2 - t_1)\gamma(1+\epsilon)}{\lambda_j}. \tag{34}$$

■

The proof of Lemma 2 is thus complete for the over-provisioned case. For the following, we will prove Corollaries 1 to 3, respectively.

A. Proof for Corollary 1

Proof: We define UT_1 (which stands for "Uncoded Transmission") as the number of time slots in $[t_1 + 1, t_3]$ when the proposed scheme schedules an *uncoded* packet transmission for Session 1. Note that by our definitions, all those uncoded transmissions must be used to transmit $X_{1,n}$ for some $n \geq n_1(t_1)$. Similarly, we also define UT_2 as the number of time slots in $[t_1+1, t_3]$ when the proposed scheme schedules an uncoded packet transmission for Session 2 packets $X_{2,n}$ with the indices being $n \geq n_2(t_1)$. Define

$$H_{1,n} = \Big|\{t > t_1 : \text{in the beginning of time } t, \text{ the scheme schedules an uncoded transmission of } X_{1,n}\}\Big|. \tag{35}$$

Since we stop an uncoded transmission if any one of the destinations successfully receives it, we have

$$\mathsf{E}\{H_{1,n}|\mathcal{A}_{t_1}\} = \frac{1}{1-(1-p_1)(1-p_2)} = \frac{1}{p_1+p_2-p_1p_2} \tag{36}$$

for all $n \geq n_1(t_1)$. As a result, the total number of time slots to transmit the uncoded session-1 packets is

$$\mathsf{UT}_1 \geq \sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1 - 1} H_{1,i},$$

where the inequality is because uncoded session 1 packets with indices less than $n_1(t_1)+1$ or larger than $n_1(t_1)+\Delta n_1 - 1$ may also be transmitted during $[t_1 + 1, t_3]$.

Similarly, the total number of time slots to transmit the uncoded session 2 packets in time $[t_1 + 1, t_3]$ is at least

$$\mathsf{UT}_2 \geq \sum_{i=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2 - 1} H_{2,i}.$$

Since each $H_{1,i}$ and $H_{2,j}$ are of i.i.d. (conditional) geometric distribution with expectation (36), for any $\epsilon_1, \delta_1 > 0$, we can choose a sufficiently large B_1 such that if $\Delta n_1 > B_1$ and $\Delta n_2 > B_1 \frac{\lambda_1}{\lambda_2}$, then

$$\begin{aligned}&\mathsf{P}\left(\mathsf{UT}_1 + \mathsf{UT}_2 > (1-\epsilon_1)\frac{\Delta n_1 + \Delta n_2 - 2}{p_1+p_2-p_1p_2}\middle|\mathcal{A}_{t_1}\right)\\ &\geq \mathsf{P}\left(\sum_{i=1}^{\Delta n_1+\Delta n_2} H_i > (1-\epsilon_1)\frac{\Delta n_1+\Delta n_2-2}{p_1+p_2-p_1p_2}\right) > 1-\delta_1,\end{aligned} \tag{37}$$

where $\{H_i\}$ are i.i.d. geometric random variables with expectation $\frac{1}{p_2+p_2-p_1p_2}$ and (37) follows from the weak law of large numbers.

Let $O_{1,n}$ denote a Bernoulli random variable that is 1 if, when repeatedly sending $X_{1,n}$ uncodedly, it was d_2 that received $X_{1,n}$ first; $O_{1,n} = 0$, if d_1 and d_2 received $X_{1,n}$ simultaneously or d_1 received it first. Symmetrically, we define the Bernoulli random variable $O_{2,n}$ such that $O_{2,n}$ is 1 if, when repeatedly sending $X_{2,n}$ uncodedly, it was d_1 that received $X_{2,n}$ first; $O_{2,n} = 0$, if d_1 and d_2 received $X_{2,n}$ simultaneously or d_2 received it first.

When $X_{1,n}$ has been received by user 2 first and not by user 1, the BS would decide whether or not to keep transmitting this packet in the uncoded fashion until it's received by user 1, or not. We define $\mathsf{FC}_{1,n}$ (which stands for "Flip a Coin") as a Bernoulli random variable to indicate the decision result. $\mathsf{FC}_{1,n} = 1$ if the BS decides to keep transmitting this packet uncodedly until it's received by user 1; $\mathsf{FC}_{1,n} = 0$ if not. By our algorithm, $\mathsf{FC}_{1,n} = 1$ with probability $1 - \frac{\lambda_1 p_1(p_1-p_1p_2)}{\lambda_2 p_2(p_2-p_1p_2)}$, $\mathsf{FC}_{1,n} = 0$ with probability $\frac{\lambda_1 p_1(p_1-p_1p_2)}{\lambda_2 p_2(p_2-p_1p_2)}$.

To distinguish from the uncoded transmission, we name the retransmission of coding opportunity of user 1 as "Single Transmission", as the single transmission is meant for user 1 only. We define $\mathsf{ST}_{1,n}$ as

$$\mathsf{ST}_{1,n} \triangleq \Big|\{t > t_1 : \text{in time } t, \text{ coding opportunity for user 1 } X_{1,n} \text{ is transmitted until user 1 receives it.}\}\Big|, \tag{38}$$

Note that for any $i \geq n_1(t_1)$, $\mathsf{ST}_{1,n} = 0$ whenever $O_{1,n} = 0$; $\mathsf{ST}_{1,n} = 0$ whenever $O_{1,n} = 1$ and $\mathsf{FC}_{1,n} = 0$; whenever we have $O_{1,n} = 1$, and $\mathsf{FC}_{1,n} = 1$, random variable $\mathsf{ST}_{1,n}$ is geometrically distributed with successful probability p_1. As a result, $\mathsf{ST}_{1,n}$ is with expectation $\frac{p_2-p_1p_2}{p_1+p_2-p_1p_2}\Big(1 - \frac{\lambda_1 p_1(p_1-p_1p_2)}{\lambda_2 p_2(p_2-p_1p_2)}\Big)\frac{1}{p_1}$ for any $n \geq n_1(t_1)$ (recall that we have temporarily suspended "expiration"). By the weak law of large numbers, we also have for any $\delta_4 > 0$, $\epsilon_4 > 0$, there exists a

B_4 such that if $\Delta n_1 > B_4$, we have

$$\mathsf{P}\Big(\sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} \mathsf{ST}_{1,i} \le (\Delta n_1 - 1)\frac{p_2 - p_1p_2}{p_1+p_2-p_1p_2} \times \Big(1 - \frac{\lambda_1 p_1(p_1-p_1p_2)}{\lambda_2 p_2(p_2-p_1p_2)}\Big)\frac{1}{p_1}(1-\epsilon_4)\Big|\mathcal{A}_{t_1}\Big) \le \delta_4. \quad (39)$$

We now define $\mathsf{CT}_{1,n}$ as follows:

$$\mathsf{CT}_{1,n} \triangleq \big|\big\{t > t_1 : \text{in time } t, \text{ packet } X_{1,n} \text{ is mixed (coded) with some other } X_{2,n'} \text{ packets.}\big\}\big|, \quad (40)$$

where $\mathsf{CT}_{1,n}$ stands for the coded transmission for packet $X_{1,n}$. Note that for any given n, the packets $X_{1,n}$ may be sent in a coded form for several (not necessarily adjacent) time slots and each time the companying $X_{2,n'}$ may be different, i.e., different n'.

Define TCT as the total number of coded transmission in time $[t_1+1, t_3]$. We then notice the following facts: (i) In the beginning of time t_3, the scheme must either transmit an uncoded packet $X_{1,n_1(t_1)+\Delta n_1-1}$, or transmit an uncoded packet $X_{2,n_2(t_1)+\Delta n_2-1}$ and it is received by one of the destinations (that is why $n_1(t)$ changes to $n_1(t_1)+\Delta n_1$, or $n_2(t)$ changes to $n_2(t_1)+\Delta n_2$). (ii) Therefore, at the end of time t_3-1, there must have $\min(L_{10}, L_{01}) = 0$. There are no packets to be coded at the end of time t_3-1. (iii) Therefore, at the end of time t_3-1, either (a) there is no $\{X_{1,n} : n \in (n_1(t_1), n_1(t_1)+\Delta n_1 - 1]\}$ in L_{01}, or (b) there is no $\{X_{2,n} : n \in (n_2(t_1), n_2(t_1)+\Delta n_2 - 1]\}$ in L_{10}. From the above three facts, we have

$$\mathsf{TCT} = \min\Big(\sum_{i=1}^{n_1(t_1)+\Delta n_1-1} \mathsf{CT}_{1,i}, \sum_{i=1}^{n_2(t_1)+\Delta n_2-1} \mathsf{CT}_{2,i}\Big). \quad (41)$$

For the following, we will prove that for any $\epsilon_5, \delta_5 > 0$, we can choose a sufficiently large B_5 such that if $\Delta n_2 > B_5$, we have

$$\mathsf{P}\left(\mathsf{TCT} > (\Delta n_2 - 1)\left(\frac{p_1 - p_1p_2}{(p_1+p_2-p_1p_2)p_2}\right)(1-\epsilon_5)\Big|\mathcal{A}_{t_1}\right) > 1 - \delta_5. \quad (42)$$

To that end, we use the following union-bound arguments and focus on the sub-series of the summations:

$$\begin{aligned}
&\mathsf{P}\Big(\mathsf{TCT} > (\Delta n_2 - 1)\Big(\frac{p_1-p_1p_2}{(p_1+p_2-p_1p_2)p_2}\Big)(1-\epsilon_5)\Big|\mathcal{A}_{t_1}\Big) \\
&= \mathsf{P}\left(\text{Eq. (41)} > (\Delta n_2 - 1)\left(\frac{p_1-p_1p_2}{(p_1+p_2-p_1p_2)p_2}\right) \times (1-\epsilon_5)\Big|\mathcal{A}_{t_1}\right) \\
&\ge 1 - \mathsf{P}\left(\sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} \mathsf{CT}_{1,i} \le (\Delta n_2 - 1) \times \Big(\frac{p_1-p_1p_2}{(p_1+p_2-p_1p_2)p_2}\Big)(1-\epsilon_5)\Big|\mathcal{A}_{t_1}\right) \\
&\quad - \mathsf{P}\left(\sum_{i=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2-1} \mathsf{CT}_{2,i} \le (\Delta n_2 - 1) \times \Big(\frac{p_1-p_1p_2}{(p_1+p_2-p_1p_2)p_2}\Big)(1-\epsilon_5)\Big|\mathcal{A}_{t_1}\right). \quad (43)
\end{aligned}$$

Note that for any $i \ge n_1(t_1)$, $\mathsf{CT}_{1,i} = 0$ if $O_{1,i} = 0$. Further, conditioning on $O_{1,i} = 1$, we have $\mathsf{FC}_{1,n} = 0$, the random variable $\mathsf{CT}_{1,i}$ is geometrically distributed with success probability p_1. As a result, by averaging over all events, we can show that $\mathsf{CT}_{1,i}$ is with expectation $\Big(\frac{p_2-p_1p_2}{p_1+p_2-p_1p_2} \cdot \frac{\lambda_1 p_1(p_1-p_1p_2)}{\lambda_2 p_2(p_2-p_1p_2)}\frac{1}{p_1}\Big)$ for any $i \ge n_1(t_1)$ (recall that we have temporarily suspended "expiration"). The weak law of large numbers thus implies that for any $\delta_6 > 0$, there exists a B_6 such that if $\Delta n_1 > B_6$, we have

$$\mathsf{P}\left(\sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} \mathsf{CT}_{1,i} \le (\Delta n_2 - 1)\Big(\frac{p_1-p_1p_2}{(p_1+p_2-p_1p_2)p_2}\Big) \times (1-\epsilon_5)\Big|\mathcal{A}_{t_1}\right) \le \delta_6. \quad (44)$$

Conditioning on $O_{2,i} = 1$, the random variable $\mathsf{CT}_{2,i}$ is geometrically distributed with success probability p_2. (Since we assume user 1 is the leading user, there is no need to flip a coin when deciding whether to set the status of a user-2 packet to be "uncoded-Tx-only"). As a result, by averaging over all events, $\mathsf{CT}_{2,i}$ is i.i.d. with expectation $\frac{p_1-p_1p_2}{(p_1+p_2-p_1p_2)p_2}$ for any $i \ge n_1(t_1)$ (recall that we have temporarily suspended "expiration").

By the weak law of large numbers, we also have for any $\delta_7 > 0$, there exists a B_7 such that if $\Delta n_2 > B_7$, we have

$$\mathsf{P}\left(\sum_{i=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2-1} \mathsf{CT}_{2,i} \le (\Delta n_2 - 1)\Big(\frac{p_1-p_1p_2}{(p_1+p_2-p_1p_2)p_2}\Big) (1-\epsilon_5)\Big|\mathcal{A}_{t_1}\right) \le \delta_7. \quad (45)$$

Jointly (44) and (45) imply that (43) can be made arbitrarily close to one by choosing sufficiently large B_6 (Δn_1 is sufficiently large so that Δn_2 is large enough) and B_7, and by setting $B_5 = \max(B_6\frac{\lambda_1}{\lambda_2}, B_7)$. Eq. (42) is thus proven.

To summarize what we have proven thus far, we define

$$\text{Term-1} \triangleq \frac{\Delta n_1 + \Delta n_2 - 2}{p_1 + p_2 - p_1 p_2}, \tag{46}$$

$$\text{Term-2} \triangleq (\Delta n_1 - 1) \left(\frac{p_2 - p_1 p_2}{p_1 + p_2 - p_1 p_2} \cdot \Big(1 - \frac{\lambda_1 p_1 (p_1 - p_1 p_2)}{\lambda_2 p_2 (p_2 - p_1 p_2)}\Big) \frac{1}{p_1} \right), \tag{47}$$

$$\text{Term-3} \triangleq \frac{(\Delta n_2 - 1)(p_1 - p_1 p_2)}{(p_1 + p_2 - p_1 p_2) p_2}. \tag{48}$$

Our previous analyses (37), (39), and (42) prove that the following three inequalities hold with close-to-one probability: (i) $\mathsf{UT}_1 + \mathsf{UT}_2 \geq (1-\epsilon_1)$Term-1, (ii) $\sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1 - 1} \mathsf{ST}_{1,i} \geq (1-\epsilon_4)$Term-2, and (iii) $\mathsf{TCT} \geq (1-\epsilon_5)$Term-3. Further, we can prove by simple arithmetic operations that

$$\begin{aligned}
&\text{Term-1} + \text{Term-2} + \text{Term-3} \\
&= \frac{\Delta n_1 + \Delta n_2 - 2}{p_1 + p_2 - p_1 p_2} + \frac{(\Delta n_2 - 1)(p_1 - p_1 p_2)}{(p_1 + p_2 - p_1 p_2) p_2} + \\
&(\Delta n_1 - 1) \left(\frac{p_2 - p_1 p_2}{p_1 + p_2 - p_1 p_2} \cdot \Big(1 - \frac{\lambda_1 p_1 (p_1 - p_1 p_2)}{\lambda_2 p_2 (p_2 - p_1 p_2)}\Big) \frac{1}{p_1} \right) \\
&= \frac{\Delta n_1 - 1}{p_1 + p_2 - p_1 p_2} + \frac{(\Delta n_1 - 1)(p_2 - p_1 p_2)}{(p_1 + p_2 - p_1 p_2) p_1} \\
&\quad + \frac{\Delta n_2 - 1}{p_1 + p_2 - p_1 p_2} + \frac{(\Delta n_2 - 1)(p_1 - p_1 p_2)}{(p_1 + p_2 - p_1 p_2) p_2} \\
&\quad - (\Delta n_1 - 1) \frac{\lambda_1 (p_1 - p_1 p_2)}{\lambda_2 (p_1 + p_2 - p_1 p_2) p_2}
\end{aligned}$$

$$\begin{aligned}
&= \frac{\Delta n_1 - 1}{p_1} + \frac{\Delta n_2 - 1}{p_2} - (\Delta n_1 - 1) \frac{\lambda_1 (p_1 - p_1 p_2)}{\lambda_2 (p_1 + p_2 - p_1 p_2) p_2} \\
&\geq \frac{\Delta n_1 - 1}{p_1} + \Big(\frac{\Delta n_1 \lambda_1 - \lambda_1 - \lambda_2}{\lambda_2} - 1 \Big) \frac{1}{p_2} \\
&\quad - (\Delta n_1 - 1) \frac{p_1 - p_1 p_2}{(p_1 + p_2 - p_1 p_2) p_2} \frac{\lambda_1}{\lambda_2} \qquad (49) \\
&= (\Delta n_1 - 1) \Big(\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2} \Big) \lambda_1 - \frac{2}{p_2} \\
&= \left\lfloor \frac{(t_2 - t_1)}{\Big(\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2} \Big) \lambda_1} \right\rfloor \Big(\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2} \Big) \lambda_1 \qquad (50) \\
&\quad - \frac{2}{p_2} \approx t_2 - t_1, \qquad (51)
\end{aligned}$$

where (50) follows from (30), and (49) is by Lemma 1.

Since for any time slot in $[t_1 + 1, t_3]$ we either send an uncoded or a coded transmission, we must have $t_3 - t_1 = \mathsf{UT}_1 + \mathsf{UT}_2 + \sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1 - 1} \mathsf{ST}_{1,i} + \mathsf{TCT}$. As as result, we have proven that for any $\epsilon_8, \delta_8 > 0$, there exists a $B_8 > 0$ such that if $t_2 - t_1 > B_8$ (so that Δn_1 and Δn_2 are sufficiently large), we have

$$\mathsf{P}\left((t_3 - t_1) > (t_2 - t_1)(1 - \epsilon_8) \middle| \mathcal{A}_{t_1} \right) > 1 - \delta_8. \tag{52}$$

Namely, with close to one probability, the random time t_3, at the end of which $n_1(t)$ is at least $n_1(t_1) + \Delta n_1$ and $n_2(t)$ is at least $n_2(t_1) + \Delta n_2$ for the first time, is no less than $t_1 + (t_2 - t_1)(1 - \epsilon_8)$. The proof for Collory 1 is complete. ■

B. Proof for Corollary 2

Proof: By Corollary 1, for any $\epsilon_8 > 0$, with close-to-one probability we have $t_3 \geq t_1 + (t_2 - t_1)(1 - \epsilon_8)$. By the definition of the random stopping time t_3, with close-to-one probability, one of the following two statements holds at the end of time $t^* \triangleq t_1 + (t_2 - t_1)(1 - \epsilon_8)$: (i) $n_1(t^*) \leq n_1(t_1) + \Delta n_1$, or (ii) $n_2(t^*) \leq n_2(t_1) + \Delta n_2$. Therefore,

$$\begin{aligned}
&\mathsf{P}\big(n_2(t^*)(1 - \epsilon_8)) \geq n_2(t_1) + \Delta n_2 \quad \& \\
&\quad n_1(t^*)(1 - \epsilon_8)) \geq n_1(t_1) + \Delta n_1 \big| \mathcal{A}_{t_1} \big) < \delta_8. \qquad (53)
\end{aligned}$$

By Lemma 2, both the distances $|\lambda_2 n_2(t_1) - \lambda_1 n_1(t_1)|$ and $|\lambda_2 n_2(t^*) - \lambda_1 n_1(t^*)|$ are upper bounded by $\max(\lambda_1, \lambda_2)$. We can thus prove that

$$\begin{aligned}
&n_2(t^*) \leq n_2(t_1) + \Delta n_2 \qquad (54) \\
&\Rightarrow n_1(t^*) \leq n_1(t_1) + \Delta n_1 + 2 \frac{\max(\lambda_1, \lambda_2)}{\lambda_1} \qquad (55) \\
&\Rightarrow n_1(t^*) \leq n_1(t_1) + \Delta n_1 + 2 \frac{\lambda_2}{\lambda_1} + 2. \qquad (56)
\end{aligned}$$

Combining (53) and (56) we have

$$\begin{aligned}
&\mathsf{P}\big(n_1(t_1 + (t_2 - t_1)(1 - \epsilon_8)) \leq n_1(t_1) + \Delta n_1 + 2 \frac{\lambda_2}{\lambda_1} + 2 \\
&\quad \big| \mathcal{A}_{t_1} \big) > 1 - \delta_8. \qquad (57)
\end{aligned}$$

We then notice that for all $j \in \{1, 2\}$, we must have $n_1(t_2) - n_1(t^*) \leq t_2 - t^*$. The reason is that for every time slot, the register $n_1(t)$ can increase at most by 1 in the over-provisioned scenario. Since the difference between t_2 and t^* is $(t_2 - t_1)\epsilon_8$, (57) implies

$$\begin{aligned}
&\mathsf{P}\left(n_1(t_2) - n_1(t_1) \leq \Delta n_1 + 2 \frac{\lambda_2}{\lambda_1} + 2 + (t_2 - t_1)\epsilon_8 \middle| \mathcal{A}_{t_1} \right) \\
&> 1 - \delta_8. \qquad (58)
\end{aligned}$$

Further, $n_1(t_2) - n_1(t_1) \leq t_2 - t_1$ since for each time slot the register $n_1(t)$ can increase by at most one. By (58), we can upper bound the expectation of $n_1(t_2) - n_1(t_1)$:

$$\begin{aligned}
&\mathsf{E}\Big\{ n_1(t_2) - n_1(t_1) | \mathcal{A}_{t_1} \Big\} \leq \Big(\Delta n_1 + \frac{\lambda_2}{\lambda_1} + 2 + (t_2 - t_1)\epsilon_8 \Big) \\
&\quad \times (1 - \delta_8) + \delta_8 (t_2 - t_1). \qquad (59)
\end{aligned}$$

By noticing that Δn_1 is linearly proportional to $(t_2 - t_1)$ while all other terms are sub-linear (with either a ϵ or a δ coefficient), (59) thus implies that for any $\epsilon > 0$, there exists a sufficiently large B such that if $t_2 - t_1 > B$, then

$$\mathsf{E}\Big\{ n_1(t_2) - n_1(t_1) | \mathcal{A}_{t_1} \Big\} \leq \frac{(t_2 - t_1)\gamma(1 + \epsilon)}{\lambda_1}. \tag{60}$$

By similar argument, we have

$$\mathsf{E}\Big\{ n_2(t_2) - n_2(t_1) | \mathcal{A}_{t_1} \Big\} \leq \frac{(t_2 - t_1)\gamma(1 + \epsilon)}{\lambda_2}. \tag{61}$$

■

C. Proof for Corollary 3

Proof: In the above analysis, we have not considered the impact of when allowing expiration. In the following, we will include expiration back to our analysis. To that end, we first notice that we can still define $H_{1,n}$, $H_{2,n}$, $\mathsf{ST}_{1,n}$, $\mathsf{CT}_{1,n}$, $\mathsf{CT}_{2,n}$ as in (35), (38), and (40), respectively. Note that now these five random variables are no longer independently distributed as the realization of one random variable, say $H_{1,n}$, may affect the distribution of the other random variables, say $\mathsf{CT}_{2,n'}$, due to expiration. Define a set of *shadow random variables* $\tilde{H}_{1,n}$, $\tilde{H}_{2,n}$, $\tilde{\mathsf{ST}}_{1,n}$, $\tilde{\mathsf{CT}}_{1,n}$, $\tilde{\mathsf{CT}}_{2,n}$ that characterize the behaviors when there is no expriation involved. More specifically, we choose $\tilde{H}_{1,n} = H_{1,n}$ if $H_{1,n}$ stops "growing" due to the $X_{1,n}$ packet being received by one of the two destinations. If $H_{1,n}$ stops growing due to the expiration of $X_{1,n}$, then we let $\tilde{H}_{1,n}$ continue to grow as an independent geometric random variable with success probability $(p_1 + p_2 - p_1p_2)$. In this way, $\tilde{H}_{1,n}$ mimics the behavior of a system with no expiration and $\tilde{H}_{1,n}$ is independent from all other random variables. In the same manner, we choose $\tilde{\mathsf{ST}}_{1,n} = \mathsf{ST}_{1,n}$ if $\mathsf{ST}_{1,n}$ stops growing due to the single transmission involving $X_{1,n}$ being received by d_1, and we let $\tilde{\mathsf{ST}}_{1,n}$ keep growing if $\mathsf{ST}_{1,n}$ stops growing due to the expiration of $X_{1,n}$. Similarly, we choose $\tilde{\mathsf{CT}}_{1,n} = \mathsf{CT}_{1,n}$ if $\mathsf{CT}_{1,n}$ stops growing due to the mixed coded transmission involving $X_{1,n}$ being received by d_1. If $\mathsf{CT}_{1,n}$ stops growing due to the expiration of $X_{1,n}$, then we let $\tilde{\mathsf{CT}}_{1,n}$ continue to grow as an independent geometric random variable. In this way, $\tilde{\mathsf{CT}}_{1,n}$ mimics the behavior of a system with no expiration and $\tilde{\mathsf{CT}}_{1,n}$ is independent from all other random variables.

Then we need to prove the following version of (52): For any $\epsilon_8, \delta_8 > 0$, there exists a sufficiently large B_8 such that for any $t_2 - t_1 > B_8$, we have

$$\delta_8 \geq \mathsf{P}\Bigg(\mathsf{UT}_1 + \mathsf{UT}_2 + \sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} \mathsf{ST}_{1,i} + \mathsf{TCT} \leq (t_2 - t_1)(1-\epsilon_8)|\mathcal{A}_{t_1}\Bigg) \tag{62}$$

$$= \mathsf{P}\Bigg(\sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} H_{1,i} + \sum_{j=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2-1} H_{2,j} + \sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} \mathsf{ST}_{1,i} + \min\Big(\sum_{k=1}^{n_1(t_1)+\Delta n_1-1} \mathsf{CT}_{1,k}, \sum_{l=1}^{n_2(t_1)+\Delta n_2-1} \mathsf{CT}_{2,l}\Big) \leq (t_2-t_1)(1-\epsilon_8)\Big|\mathcal{A}_{t_1}\Bigg) \tag{63}$$

Note that conditioning on the event $\mathcal{A}_{t_1}$ (see the definition of $\mathcal{A}_{t_1}$ in Appendix A), during time $(t_1, t_1 + (t_2 - t_1)(1-\epsilon_8)]$, no packets with indices $\geq n_1(t_1)$ for session 1 and packets with indices $\geq n_2(t_1)$ for session 2 will expire. Therefore, conditioning on $\mathcal{A}_{t_1}$ any realization of $H_{1,i}$, $H_{2,j}$, $\mathsf{ST}_{1,i}$, $\mathsf{CT}_{1,k}$, and $\mathsf{CT}_{2,l}$ in (63) must not result in any expiration for packets with indices $\geq n_1(t_1)$ for session 1 and packets with indices $\geq n_2(t_1)$ for session 2. As a result, we have

$$\mathsf{P}\Bigg(\sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} H_{1,i} + \sum_{j=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2-1} H_{2,j} + \min\Bigg(\sum_{k=1}^{n_1(t_1)+\Delta n_1-1} \mathsf{CT}_{1,k}, \sum_{l=1}^{n_2(t_1)+\Delta n_2-1} \mathsf{CT}_{2,l}\Bigg) + \sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} \mathsf{ST}_{1,i} \leq (t_2-t_1)(1-\epsilon_8)\Bigg|\mathcal{A}_{t_1}\Bigg)$$

$$= \mathsf{P}\Bigg(\sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} \tilde{H}_{1,i} + \sum_{j=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2-1} \tilde{H}_{2,j} + \min\Bigg(\sum_{k=n_1(t_1)+1}^{n_1(t_1)+\Delta n_2-1} \tilde{\mathsf{CT}}_{1,k}, \sum_{l=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2-1} \tilde{\mathsf{CT}}_{2,l}\Bigg) + \sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} \tilde{\mathsf{ST}}_{1,i} \leq (t_2-t_1)(1-\epsilon_8)\Bigg|\mathcal{A}_{t_1}\Bigg), \tag{64}$$

because for those realizations, the probability distributions of the shadow random variables and the actual random variables are the same for those packets that are with indices $\geq n_1(t_1)$ for session 1 or that are with indices $\geq n_2(t_1)$ for session 2, and that are transmitted between $[t_1+1, t_2]$. Since (52) holds for the case without expiration, (64) is smaller than δ_8 with sufficiently large B_8. (63) is thus proven. We can then follow the same analysis as in (52) to (60). ■

We have shown the case when user 1 is the leading user. By the same approach, we can also show similar results for the case with user 2 as the leading user (that is, $\frac{\lambda_1 p_1(p_1-p_1p_2)}{\lambda_2 p_2(p_2-p_1p_2)} > 1$, and $\frac{1}{\gamma} = \frac{1/\lambda_1}{p_1+p_2-p_1p_2} + \frac{1/\lambda_2}{p_2}$). Then the proof for the over-provisioned case of Lemma 2 is complete.

Appendix B
Proof for The Under Provisioned Case of Lemma 2

Proof: The proof for the under-provisioned case ($\gamma < 1$) of Lemma 2 is similar. The goal is to show that for any $\epsilon > 0$, there exists a $B > 0$ such that for all fixed t_1 and t_2 satisfying $(t_2 - t_1) = B$, we have for $j = 1, 2$,

$$\mathsf{E}\Big\{n_j(t_2) - n_j(t_1)\Big|t_2 < \min(\lambda_1 n_1(t_1), \lambda_2 n_2(t_1))\Big\} \leq \frac{(t_2-t_1)(1+\epsilon)}{\lambda_j}. \tag{65}$$

Define Δn_1 and Δn_2 the same way as in (30) and (31). Define t_3 as the first (random) time slot for which in the end of time t_3, the BS has scheduled transmission for at least Δn_1 uncoded packets for session 1 and Δn_2 uncoded packets for session 2, respectively. Note that since we are dealing with the under-provisioned case, some packets are dropped and will never be transmitted. The way we define t_3 here is to count only those Δn_1 and Δn_2 uncoded packets that are actually transmitted. (Note that for the over-provisioned case when we do not drop any packets, the above t_3 definition is identical

to the one used in the proof of Corollary 1.) We then relabel the next Δn_1 packets including packet $n_1(t_1)$ (that have been transmitted by the BS) from session 1, as $\overline{n}_1(t_1)$, ..., $\overline{n}_1(t_1)+\Delta n_1-1$. We also relabel the next Δn_2 packets (that have been transmitted by the BS) including packet $n_2(t_1)$ from session 2, as $\overline{n}_2(t_1)$, ..., $\overline{n}_2(t_1)+\Delta n_2-1$.

We first examine how long it takes before the BS finishes transmitting packets $\overline{n}_1(t_1)$, ..., $\overline{n}_1(t_1)+\Delta n_1-1$ for session 1, and finishes transmitting packets $\overline{n}_2(t_1)$, ..., $\overline{n}_2(t_1)+\Delta n_2-1$ for session 2. That is, we want to understand the distribution of the random stopping time t_3. We then would examine at the end of time t_3, how would indices $n_1(t_3)$ and $n_2(t_3)$ be. That is, we want to investigate how many packets (out of $n_j(t_3)-n_j(t_1)$ packets) for each session have been transmitted by the BS or how many of them (out of $n_j(t_3)-n_j(t_1)$ packets) were discarded without transmission due to congestion control in Lines 2 to 16 of the IDNC scheme.

We can apply a similar proof and show that with close-to-one probability t_3 is no less than $t_1+(t_2-t_1)(1-\epsilon')$. Namely, at most Δn_1 (resp. Δn_2) uncoded packets have been transmitted for session 1 (resp. session 2) by the end of time $t_1+(t_2-t_1)(1-\epsilon')$.

By our congestion control mechanism (Lines 2 to 16), whenever x_1 is increased by 1, then the BS would schedule one more uncoded packet of session 1 to be transmitted. Recall that in the under-provisioned case, $\gamma<1$, that is $\frac{1/\lambda_1}{p_1}+\frac{1/\lambda_2}{p_1+p_2-p_1p_2}>1$. Hence after the BS finishes transmitting Δn_1 uncoded packets from session 1, the register n_1 is at most increased by

$$\Delta n_1\left(\frac{1/\lambda_1}{p_1}+\frac{1/\lambda_2}{p_1+p_2-p_1p_2}\right).$$

Using the definition of Δn_1 in (30), we then have $\Delta n_1(\frac{1/\lambda_1}{p_1}+\frac{1/\lambda_2}{p_1+p_2-p_1p_2})\approx\frac{t_2-t_1}{\lambda_1}$. Similarly, the register n_2 is at most increased by

$$\Delta n_1\left(\frac{1/\lambda_1}{p_1}+\frac{1/\lambda_2}{p_1+p_2-p_1p_2}\right)\frac{\lambda_1}{\lambda_2}\approx\frac{t_2-t_1}{\lambda_2}.$$

Combining the above observations together, and following a similar proof, $n_j(t_2)-n_j(t_1)$, the increment of the actual indices, must satisfy

$$\mathsf{E}\Big\{n_j(t_2)-n_j(t_1)|\mathcal{A}_{t_1}\Big\}\leq\frac{(t_2-t_1)(1+\epsilon)}{\lambda_j}. \tag{66}$$

The critical-provisioned case can be proven in the similar way. Note that, for the critical-provisioned case, $\gamma=\frac{1/\lambda_1}{p_1}+\frac{1/\lambda_2}{p_1+p_2-p_1p_2}=1$. The proof is thus complete.

■