

10-3-2011

# Optimal Immediately-Decodable Inter-session Network Coding (IDNC) Schemes for Two Unicast Sessions with Hard Deadline Constraints

Xiaohang Li

*Center for Wireless Systems and Applications, Electrical and Computer Engineering, Purdue University, li179@purdue.edu*

Chih-Chun Wang

*Center for Wireless Systems and Applications, Electrical and Computer Engineering, Purdue University, chihw@purdue.edu*

Xiaojun Lin

*Purdue University, linx@ecn.purdue.edu*

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

Li, Xiaohang; Wang, Chih-Chun; and Lin, Xiaojun, "Optimal Immediately-Decodable Inter-session Network Coding (IDNC) Schemes for Two Unicast Sessions with Hard Deadline Constraints" (2011). *ECE Technical Reports*. Paper 423.  
<http://docs.lib.purdue.edu/ecetr/423>

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

Optimal Immediately-Decodable Inter-session Network Coding (IDNC)  
Schemes for Two Unicast Sessions with Hard Deadline Constraints

Xiaohang Li

Chih-Chun Wang

Xiaojun Lin

TR-ECE-11-17

October 3, 2011

School of Electrical and Computer Engineering

1285 Electrical Engineering Building

Purdue University

West Lafayette, IN 47907-1285

# Optimal Immediately-Decodable Inter-Session Network Coding (IDNC) Schemes for Two Unicast Sessions with Hard Deadline Constraints

Xiaohang Li, Chih-Chun Wang, and Xiaojun Lin

**Abstract**—In this paper, we study inter-session network coding for sending two unicast sessions over an unreliable wireless channel. Each unicast session transmits a stored video file, whose packets have hard sequential deadline constraints. We first characterize the capacity region (with inter-session network coding) for the transmission rates of the two unicast sessions under heterogeneous channel conditions and heterogeneous deadline constraints. We then develop immediately-decodable network coding (IDNC) schemes for controlling packet transmissions for the unicast sessions subject to hard deadline constraints. In contrast to our prior work that focuses 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 deadlines. Nonetheless, we establish the asymptotic optimality of the proposed IDNC scheme when the file sizes are sufficiently large.

## I. INTRODUCTION

The advance of broadband wireless technologies has enabled a number of innovative wireless services. It is now common to use multimedia services in 3G/4G cellular networks or WiFi, 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 study inter-session network coding for 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 at the same time. 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 will focus on using inter-session network coding (NC) to improve the deadline-constrained streaming throughput in a single-cell.

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. 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, [7]–[9] discuss how to use NC to control the inter-session network traffic, but consider lossless channels only. [10] 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 do not focus on delay. In contrast, our paper focuses on the delay aspect of multiple unicast sessions. Readers are referred to [6], [15]–[19] and the references therein for the delay analysis in the simpler setting of a single multicast/broadcast session.

To combat the delay inefficiency of NC, recent practical protocols have focused more on the “immediately decodable” NC (IDNC) schemes [10], [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, and suppose that  $d_1$  has overheard  $Y$  and  $d_2$  has overheard  $X$ . By exploiting this mismatch of reception, the BS can 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 substantially smaller decoding delay, and incur much lower encoding complexity since only binary field is used. As a

X. Li, C.-C. Wang and X. Lin are with Center for Wireless Systems and Applications, School of ECE, Purdue University, West Lafayette, IN, 47907. Email: {li179, chihw}@purdue.edu, linx@ecn.purdue.edu

result, IDNC schemes generally demonstrate much faster startup phase [21], and is more suitable for time-sensitive applications.

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 II-B). 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. 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 [22], 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 describes the IDNC schemes for deadline-constrained streaming. Section IV provides the throughput analysis of IDNC schemes under heterogeneous deadline constraints and heterogeneous channel conditions, which is the main contribution of this paper. Section V presents the simulation results for the proposed IDNC schemes. Section VI 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  $j = 1, 2$

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

where  $\lambda_j$  is the (sequential) deadline increment for session  $j$ . In this work, we consider the heterogeneous deadlines, i.e.

$\lambda_1$  and  $\lambda_2$  may be different. We assume that  $\lambda_1 N_1 = \lambda_2 N_2$ , that is, the total display time for each video file is the same<sup>1</sup>.

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.

If coding is not allowed, the BS can only transmit uncoded packets. Suppose packet  $X_{1,n}$  is transmitted at time  $t$ , and user 1 does not receive it. After receiving the feedback at the end of time  $t$ , the BS may decide to retransmit  $X_{1,n}$ ; or may decide to move to the next packet  $X_{1,n+1}$ ; or may decide to send packet  $X_{2,n'}$  for the other session instead. In one slot, the BS can add a set of unexpired packets together and send the resultant coded packet to all users. We say that (unexpired) packet  $X_{1,n}$  is a (potential) *coding opportunity* involving user 1 when packet  $X_{1,n}$  has been received by user 2 but not by user 1. Symmetrically, (unexpired) packet  $X_{2,n}$  is a (potential) *coding opportunity* involving user 2 when packet  $X_{2,n}$  has been received by user 1 but not by user 2. A coding opportunity of user 1 can be combined with a coding opportunity of user 2 to form a coded packet. 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 normalized throughput, defined as  $\min\left(\frac{E\{N_1^{\text{success}}\}}{N_1}, \frac{E\{N_2^{\text{success}}\}}{N_2}\right)$ .

### A. The Capacity Region

Consider an interval  $(0, T]$ . Suppose that during this interval,  $r_1 T$  packets from session 1 must be delivered. Hence, transmitting those packets (either in an uncoded or a coded way) would require  $\frac{r_1 T}{p_1}$  number of time slots on average. In the same interval  $(0, T]$ , suppose  $r_2 T$  packets from session 2 must be delivered. Further, suppose that on average session 1 has more coding opportunities than that could be combined with session 2’s coding opportunities. Note that even though

<sup>1</sup>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 since there is no other session to be coded with.

sometimes we may use NC to serve two destinations simultaneously, before doing so each session-2 packet needs to be first transmitted uncodedly until it is received by at least one of the destinations. Note that if this uncoded transmission is received by  $d_2$ , then no further transmission of this packet is needed. If the uncoded transmission is received by  $d_1$ , then the following coded transmission is already counted in the term  $\frac{r_1 T}{p_1}$ . As a result, the uncoded transmission of session-2 packets only takes  $\frac{r_2 T}{1 - (1-p_1)(1-p_2)}$  number of time slots on average. In this case, since the time slots used to convey session-1 packets (either uncodedly or codedly) must be disjoint from the time slots to transmit session-2 uncodedly, we must have

$$\begin{aligned} \frac{r_1 T}{p_1} + \frac{r_2 T}{p_1 + p_2 - p_1 p_2} &\leq T \\ \Leftrightarrow \frac{r_1}{p_1} + \frac{r_2}{p_1 + p_2 - p_1 p_2} &\leq 1. \end{aligned} \quad (1)$$

By swapping the roles of sessions 1 and 2, we also have

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

The work in [14] has shown that (1) and (2) together describe the exact capacity region in a classic throughput-based setting without hard deadline constraints. Note that the capacity without deadlines is always an upper bound of the capacity with deadlines. Moreover, we also notice that when  $T = \lambda_1 N_1 = \lambda_2 N_2$ , the best possible performance of a deadline-constrained system is to successfully send  $N_1$  and  $N_2$  packets, respectively, while respecting the deadlines. Therefore, the maximum possible effective-rate of a deadline-constrained system becomes  $(\frac{1}{\lambda_1}, \frac{1}{\lambda_2})$ . By combining the above two observations, one can easily prove the following upper bound:

*Proposition 1:* For any scheme in a deadline constrained system, the expected achievable throughput vector  $(\frac{\mathbb{E}\{N_1^{\text{success}}\}}{\lambda_1 N_1}, \frac{\mathbb{E}\{N_2^{\text{success}}\}}{\lambda_2 N_2})$ , defined in Section II, must be in the following region:

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

### B. The Challenges When Designing NC Schemes For Deadline-Constrained Systems

To motivate our design choices, we describe in the following a couple of challenges that will arise when designing an IDNC scheme. We refer to the simpler setting of a single multicast session with homogeneous channel ( $p_1 = p_2$ ) and homogeneous deadlines, which has been studied in [6]. Namely, both destinations  $d_1$  and  $d_2$  are interested in the packets of the same video file (in contrast with the setting of this work in which each  $d_j$  is requesting different packets  $X_{j,n}$ ). For the single-multicast setting, the following IDNC scheme turns out to be optimal even with deadline constraints [6]: Whenever there coexist two coding-opportunities that could be combined (one for  $d_1$  and one for  $d_2$ ), we mix

packets together and send a coded packet. Whenever there are no coding opportunities that can be combined, among those packets that have never been heard by any destination, we choose the oldest one and keep sending it until it is heard by at least one destination. If this uncoded packet is received by the intended user, we move to the next packet. Otherwise, it becomes a new coding opportunity. We then check again whether there are coding opportunities that could be combined, and so on. For the following discussion, we refer to the above scheme as “the simple IDNC scheme.”

In the multiple-unicast setting, one might expect that the simple IDNC scheme will also achieve the optimal capacity region in (3). However, this is not true and the behavior is quite different. Again consider the setting of homogeneous channel  $p_1 = p_2$  and homogeneous deadlines  $\lambda_1 = \lambda_2$  but with two coexisting sessions,  $\{X_{1,n} : \forall n\}$  and  $\{X_{2,n} : \forall n\}$ . 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 (1) or (2), it is simply impossible to meet the deadlines of all packets. We call this the *under-provisioned* scenario.<sup>2</sup> As we will explain below, the simple IDNC scheme is strictly suboptimal for the 2-unicast setting considered in this work. The intuition behind this performance loss is the following. In the 2-unicast setting, each uncoded transmission, say sending  $X_{j,n}$ , can only benefit one particular user,  $d_j$ , while sending a code packet  $[X_{1,n_1} + X_{2,n_2}]$  can benefit both users. Since sending a coded packet achieves higher throughput, an optimal scheme needs to send as many coded packets as possible. On the other hand, in the simple IDNC scheme, a packet, say  $X_{1,n}$ , needs to be overheard by the other user  $d_2$  (when we send  $X_{1,n}$  uncodedly) before it can participate in coded transmission. In other words, a coded transmission involving  $X_{1,n}$  can happen only after we have sent  $X_{1,n}$  uncodedly first. As a result, in terms of the “life cycle” of a given packet, the first half of the life cycle is when the packet has not been heard by any destination (thus is ready to be sent uncodedly), and the second half of the life cycle is when the packet has been overheard by the other destination (thus is ready to be sent codedly). This causality relationship, i.e., sending uncoded  $X_{1,n}$  (first half of the life cycle) before sending a coded packet involving  $X_{1,n}$  (second half of the life cycle), causes a new problem in the under-provisioned scenario. More specifically, since we have a deadline for each packet and the system is under-provisioned, the packet is likely to expire. Therefore, most packet expiration will happen when a packet is waiting to be sent codedly. We thus will not have as many coded transmission as we would have hoped for. Note that this problem does not arise in multicast with deadlines [6], because there, an uncoded transmission benefits as many users as a coded transmission. To solve this problem, we propose to *actively drop a certain number of packets in advance*. By deliberately discarding some packets we relax

<sup>2</sup>The concept of under provisioning can be defined similarly for the single-multicast setting.

the deadlines for those not-discarded packets. Therefore, those not-discarded packets are less likely to expire, and can stay as “coding-ready” phase (the second halves of their life cycles) longer. More coded packets will thus be transmitted, and the throughput will improve.

The second challenge arises from the heterogeneity of the channel and the deadlines, which is orthogonal from the previous under-provisioned scenario. Suppose that the system is over-provisioned, i.e., the rate vector  $(1/\lambda_1, 1/\lambda_2)$  satisfies both (1) and (2). Due to the heterogeneity of the channels and deadlines, one user, say user 1, may on average have more coding opportunities than that could be combined with user 2’s coding opportunities. In the simple IDNC scheme, when there is no matched coding opportunity, the BS will keep transmitting new uncoded packets. As a result, user-1’s outstanding coding opportunities will likely to expire, which will reduce the throughput. In contrast, *the optimal solution is to retransmit those coding opportunities of user 1 uncodedly* rather than waiting for future coding opportunities to come.

To make the above discussion rigorous, 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 (1) and (2). Recall that in (1),  $\frac{r_1 T}{p_1}$  counts all the uncoded and coded transmissions combined for conveying  $r_1 T$  session-1 packets. Similarly, in (2),  $\frac{r_1 T}{p_1 + p_2 - p_1 p_2}$  counts all the uncoded transmissions of session-1 packets. As a result, the difference of the two is the number of coded transmissions for session 1. By the same argument, the difference between  $\frac{r_2 T}{p_1 + p_2 - p_1 p_2}$  and  $\frac{r_2 T}{p_2}$  is the number of coded transmissions for session 2. If we have

$$\begin{aligned} \frac{r_1 T}{p_1} - \frac{r_1 T}{p_1 + p_2 - p_1 p_2} &> \frac{r_2 T}{p_2} - \frac{r_2 T}{p_1 + p_2 - p_1 p_2} \\ \Leftrightarrow \frac{\lambda_1 p_1 (p_1 - p_1 p_2)}{\lambda_2 p_2 (p_2 - p_1 p_2)} &< 1, \end{aligned} \quad (4)$$

then from our previous arguments, we will have much more user-1 coding opportunities than those of user 2. As a result, some user-1 coding opportunities may expire even before being coded together with the user-2 packets. To recover this sub-optimality, when (4) is satisfied, an optimal IDNC scheme should continue sending some user-1 packet in an uncoded way even after it has been overheard by user 2. For future reference, we say “user 1 is a leading user” if (4) is satisfied since user-1 now has more coding opportunities than that could be combined with user-2’s coding opportunities. In the next section, 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.

### III. THE SCHEME

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  $\text{status1}[i]$ ,  $i = 1, \dots, N_1$ , and  $\text{status2}[i]$ ,  $i = 1, \dots, N_2$  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 in 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 same time slot.

In the following, we present our IDNC scheme. In the time origin, the BS first initializes the following variables:  $n_1 \leftarrow 1$ ,  $n_2 \leftarrow 1$ ,  $L_{10} \leftarrow \emptyset$ ,  $L_{01} \leftarrow \emptyset$ ,  $\text{status1}[i] \leftarrow \text{not-processed}$ ,  $\text{status2}[i] \leftarrow \text{not-processed}$ , for all  $i$ ;  $x_1, y_1, x_2, y_2 \leftarrow 0$ . For convenience, we use  $\gamma$  to denote a constant value used throughout the algorithm, which can be easily computed by the BS. That is,

$$\gamma \triangleq \min \left( \frac{1}{\frac{1}{\lambda_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}}, \frac{1}{\frac{1/\lambda_1}{p_1 + p_2 - p_1 p_2} + \frac{1/\lambda_2}{p_2}} \right). \quad (5)$$

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**  $\text{status1}[n_1] = \text{not-processed}$  **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**  $a \geq 1 - \frac{N_2(\frac{p_1}{p_2} - p_1)}{N_1(\frac{p_2}{p_1} - p_2)}$  **then**
- 8:           $\text{status1}[n_1] \leftarrow \text{coding-eligible}$
- 9:       **else**
- 10:           $\text{status1}[n_1] \leftarrow \text{uncoded-Tx-only}$
- 11:       **end if**
- 12:     **else**
- 13:        $\text{status1}[n_1] \leftarrow \text{dropped}$
- 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 linear
      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 1 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 linear 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 end of the transmission. When we reach the end 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, 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 find an “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  $\gamma$  is indeed to decide whether the system is over-provisioned ( $\gamma \geq 1$ ) or under-provisioned ( $\gamma < 1$ ). As explained in Section II-B, we drop a packet only when  $\gamma < 1$ , and Lines 3 to 5 decide the optimal packet dropping ratio. 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 transmitted, as explained in Section II-B, we sometimes need to forcefully send packets in an uncoded form for the “leading user”. If user 1 is the leading user, then  $\frac{\lambda_1 p_1 (p_1 - p_1 p_2)}{\lambda_2 p_2 (p_2 - p_1 p_2)} < 1$ . 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  $\frac{\lambda_1 p_1 (p_1 - p_1 p_2)}{\lambda_2 p_2 (p_2 - p_1 p_2)} > 1$  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  $n_2\lambda_2$ , see 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 1-3), 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 way. 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 be transmitted again in the form of uncoded packet. 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.

## IV. MAIN RESULT: PERFORMANCE ANALYSIS OF THE NEW IDNC SCHEME

The performance of the proposed new IDNC scheme is characterized as follows.

*Proposition 2:* For any given system parameters  $p_1$ ,  $p_2$ ,  $\lambda_1$ , and  $\lambda_2$ , let  $\beta^*$  denote the largest  $\beta$  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 (1) and (2). 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  $\mathbb{E}\{N_1^{\text{success}}\}/N_1 \geq \frac{\beta^*}{\lambda_1} - \epsilon$  and  $\mathbb{E}\{N_2^{\text{success}}\}/N_2 \geq \frac{\beta^*}{\lambda_2} - \epsilon$ .

Proposition 2 shows that our IDNC scheme achieves 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  $\lambda_1, \lambda_2, p_1$ , and  $p_2$ . Then 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  $j = 1, 2$ ,

$$\begin{aligned} & \mathbb{E} \left\{ n_j(t_2) - n_j(t_1) \mid t_2 < \min(\lambda_1 n_1(t_1), \lambda_2 n_2(t_1)) \right\} \\ & \leq \frac{(t_2 - t_1) \max(\gamma, 1)(1 + \epsilon)}{\lambda_j}. \end{aligned} \quad (6)$$

The high-level intuition of this lemma is provided as follows. Consider any two fixed time instants  $t_1$  and  $t_2$ . 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  $\frac{\max(\gamma, 1)(t_2 - t_1)}{\lambda_1}$ . In other words, the growth of  $n_1(t)$  is proportional to how fast the packets of session 1 expire. Also 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  $X_{1, n_1(t_1)}, X_{1, n_1(t_1)+1}, \dots, X_{1, n_1(t_2)-1}$  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  $t_2 - t_1$  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 (6). Next we present a detailed proof of Lemma 2 for the over-provisioned case, that is,  $\gamma \geq 1$ .

*Proof:* We first discuss the case that user 1 is the leading user, that is,  $\frac{\lambda_1 p_1 (p_1 - p_1 p_2)}{\lambda_2 p_2 (p_2 - p_1 p_2)} < 1$ . So  $\frac{1}{\gamma} = \frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}$ . The following discussion is conditioned on the event that in the end of time  $t_1$ , we have  $\mathcal{A}_{t_1} \triangleq \{t_2 < \lambda_1 n_1(t_1), t_2 < \lambda_2 n_2(t_1)\}$ . Define

$$\begin{aligned} \Delta n_1 &= \left\lfloor \frac{(t_2 - t_1)}{\left( \frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2} \right) \lambda_1} \right\rfloor + 1, \\ \Delta n_2 &= \left\lfloor \frac{(t_2 - t_1)}{\left( \frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2} \right) \lambda_2} \right\rfloor + 1. \end{aligned}$$

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 for which in the end of time  $t_3$ , both  $n_1(t)$  is at least  $n_1(t_1) + \Delta n_1$  and  $n_2(t)$  is at least  $n_2(t_1) + \Delta n_2$  for the first time.

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

$$H_{1, n} = \left\{ t > t_1 : \text{in the beginning of time } t, \text{ the scheme schedules an uncoded transmission of } X_{1, n} \right\}. \quad (7)$$

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

$$\mathbb{E}\{H_{1, n} | \mathcal{A}_{t_1}\} = \frac{1}{1 - (1 - p_1)(1 - p_2)} = \frac{1}{p_1 + p_2 - p_1 p_2} \quad (8)$$

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

$$UT_1 \geq \sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} H_{1, i},$$

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

$$UT_2 \geq \sum_{i=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2-1} H_{2, i}.$$

Since each  $H_{1, i}$  and  $H_{2, j}$  are i.i.d. (conditional) geometric distribution with expectation (8), 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

$$\begin{aligned} & \mathbb{P} \left( UT_1 + UT_2 > (1 - \epsilon_1) \frac{\Delta n_1 + \Delta n_2 - 2}{p_1 + p_2 - p_1 p_2} \mid \mathcal{A}_{t_1} \right) \\ &= \mathbb{P} \left( \sum_{i=1}^{\Delta n_1 + \Delta n_2} H_i > (1 - \epsilon_1) \frac{\Delta n_1 + \Delta n_2 - 2}{p_1 + p_2 - p_1 p_2} \right) > 1 - \delta_1, \end{aligned} \quad (9)$$

where  $\{H_i\}$  are i.i.d. geometric random variables with expectation  $\frac{1}{p_2 + p_2 - p_1 p_2}$  and (9) follows from the weak law of large numbers.

Let  $O_{1, n}$  denote a Bernoulli random variable that is 1 if when 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 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 to keep transmitting this packet in the uncoded fashion until it's received by user 1, or not. We define  $FC_{1,n}$  (which stands for "Flip a Coin") as a Bernoulli random variable to indicate the decision result.  $FC_{1,n} = 1$  if the BS decides to keep transmitting this packet uncodedly until it's received by user 1;  $FC_{1,n} = 0$  if not. By our algorithm,  $FC_{1,n} = 1$  with probability  $1 - \frac{N_2(\frac{p_1}{p_2} - p_1)}{N_1(\frac{p_2}{p_1} - p_2)}$ ,  $FC_{1,n} = 0$  with probability  $\frac{N_2(\frac{p_1}{p_2} - p_1)}{N_1(\frac{p_2}{p_1} - p_2)}$ .

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  $ST_{1,n}$  as

$$ST_{1,n} \triangleq \left\{ t > t_1 : \text{in time } t, \text{ coding opportunity for user 1 } X_{1,n} \text{ is transmitted until user 1 receives it.} \right\}, \quad (10)$$

Note that for any  $i \geq n_1(t_1)$ ,  $ST_{1,n} = 0$  whenever  $O_{1,n} = 0$ ;  $ST_{1,n} = 0$  whenever  $O_{1,n} = 1$  and  $FC_{1,n} = 0$ ; whenever we have  $O_{1,n} = 1$ , and  $FC_{1,n} = 1$ , random variable  $ST_{1,n}$  is geometrically distributed with successful probability  $p_1$ . As a result,  $ST_{1,n}$  is with expectation  $\frac{p_2 - p_1 p_2}{p_1 + p_2 - p_1 p_2} \left( 1 - \frac{N_2(p_1 - p_1 p_2) \frac{p_1}{p_2}}{N_1(p_2 - p_1 p_2)} \right) \frac{1}{p_1}$  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

$$\begin{aligned} & \mathbb{P} \left( \sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} ST_{1,i} \leq (\Delta n_1 - 1) \frac{p_2 - p_1 p_2}{p_1 + p_2 - p_1 p_2} \right. \\ & \quad \left. \times \left( 1 - \frac{N_2(p_1 - p_1 p_2) \frac{p_1}{p_2}}{N_1(p_2 - p_1 p_2)} \right) \frac{1}{p_1} (1 - \epsilon_4) \middle| \mathcal{A}_{t_1} \right) \leq \delta_4. \end{aligned} \quad (11)$$

We now define  $CT_{1,n}$  as follows:

$$CT_{1,n} \triangleq \left\{ t > t_1 : \text{in time } t, \text{ packet } X_{1,n} \text{ is mixed (coded) with some other } X_{2,n'} \text{ packets.} \right\}, \quad (12)$$

where  $CT_{1,n}$  stands for the coded transmission for packet  $X_{1,n}$ . 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  $\min(L_{10}, L_{01}) = 0$ . That are no packets to be coded in 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

$$TCT = \min \left( \sum_{i=1}^{n_1(t_1)+\Delta n_1-1} CT_{1,i}, \sum_{i=1}^{n_2(t_1)+\Delta n_2-1} CT_{2,i} \right). \quad (13)$$

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

$$\mathbb{P} \left( TCT > (\Delta n_2 - 1) \left( \frac{p_1 - p_1 p_2}{(p_1 + p_2 - p_1 p_2) p_2} \right) (1 - \epsilon_5) \middle| \mathcal{A}_{t_1} \right) > 1 - \delta_5. \quad (14)$$

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

$$\begin{aligned} & \mathbb{P} \left( TCT > (\Delta n_2 - 1) \left( \frac{p_1 - p_1 p_2}{(p_1 + p_2 - p_1 p_2) p_2} \right) (1 - \epsilon_5) \middle| \mathcal{A}_{t_1} \right) \\ &= \mathbb{P} \left( \text{Eq. (13)} > (\Delta n_2 - 1) \left( \frac{p_1 - p_1 p_2}{(p_1 + p_2 - p_1 p_2) p_2} \right) \right. \\ & \quad \left. \times (1 - \epsilon_5) \middle| \mathcal{A}_{t_1} \right) \\ &\geq 1 - \mathbb{P} \left( \sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} CT_{1,i} \leq (\Delta n_2 - 1) \right. \\ & \quad \left. \times \left( \frac{p_1 - p_1 p_2}{(p_1 + p_2 - p_1 p_2) p_2} \right) (1 - \epsilon_5) \middle| \mathcal{A}_{t_1} \right) \\ & \quad - \mathbb{P} \left( \sum_{i=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2-1} CT_{2,i} \leq (\Delta n_2 - 1) \right. \\ & \quad \left. \times \left( \frac{p_1 - p_1 p_2}{(p_1 + p_2 - p_1 p_2) p_2} \right) (1 - \epsilon_5) \middle| \mathcal{A}_{t_1} \right). \end{aligned} \quad (15)$$

Note that for any  $i \geq n_1(t_1)$ ,  $CT_{1,i} = 0$  if  $O_{1,i} = 0$ , and conditioning on  $O_{1,i} = 1$ ,  $FC_{1,n} = 0$ , the random variable  $CT_{1,i}$  is geometrically distributed with success probability  $p_1$ . Moreover,  $CT_{1,i}$  is with expectation  $\left( \frac{p_2 - p_1 p_2}{p_1 + p_2 - p_1 p_2} \cdot \frac{N_2(\frac{p_1}{p_2} - p_1)}{N_1(\frac{p_2}{p_1} - p_2)} \frac{1}{p_1} \right)$  for any  $i \geq n_1(t_1)$  (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

$$\mathbb{P} \left( \sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} CT_{1,i} \leq (\Delta n_2 - 1) \left( \frac{p_1 - p_1 p_2}{(p_1 + p_2 - p_1 p_2) p_2} \right) \right. \\ \left. \times (1 - \epsilon_5) \middle| \mathcal{A}_{t_1} \right) \leq \delta_6. \quad (16)$$

Note that for any  $i \geq n_2(t_1)$ ,  $CT_{2,i} = 0$  if  $O_{2,i} = 0$ , and conditioning on  $O_{2,i} = 1$ , the random variable  $CT_{2,i}$  is geometrically distributed with success probability  $p_2$ . Moreover,  $CT_{2,i}$  is i.i.d. with expectation  $\left( \frac{p_1 - p_1 p_2}{p_1 + p_2 - p_1 p_2} \cdot \frac{1}{p_2} \right)$  for any  $i \geq n_1(t_1)$  (recall that we have temporarily suspended "expiration"). By the weak law of large numbers, we also

have for any  $\delta_7 > 0$ , there exists a  $B_7$  such that if  $\Delta n_2 > B_7$ , we have

$$\begin{aligned} & \mathbb{P} \left( \sum_{i=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2-1} \text{CT}_{2,i} \leq (\Delta n_2 - 1) \left( \frac{p_1 - p_1 p_2}{(p_1 + p_2 - p_1 p_2) p_2} \right) \right. \\ & \quad \left. (1 - \epsilon_5) \middle| \mathcal{A}_{t_1} \right) \end{aligned} \quad (17)$$

$$\begin{aligned} & = \mathbb{P} \left( \sum_{i=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2-1} \text{CT}_{2,i} \leq (\Delta n_2 - 1) \right. \\ & \quad \left. \left( \frac{p_1 - p_1 p_2}{(p_1 + p_2 - p_1 p_2) p_2} \right) (1 - \epsilon_5) \middle| \mathcal{A}_{t_1} \right) \leq \delta_7. \end{aligned} \quad (18)$$

Jointly (16) and (18) imply that (15) can be made arbitrarily close to one by choosing a sufficiently large  $B_6$  ( $\Delta n_1$  is sufficiently large so that  $\Delta n_2$  is large enough) and  $B_7$  and setting  $B_2 = \max(B_6 \frac{\lambda_1}{\lambda_2}, B_7)$ . Eq. (14) is thus proven.

By simple arithmetic operations, we have the following equations:

$$\begin{aligned} & \frac{\Delta n_1 + \Delta n_2 - 2}{p_1 + p_2 - p_1 p_2} + \frac{(\Delta n_2 - 1)(p_1 - p_1 p_2)}{(p_1 + p_2 - p_1 p_2) p_2} + \\ & (\Delta n_1 - 1) \left( \frac{p_2 - p_1 p_2}{p_1 + p_2 - p_1 p_2} \cdot \left( 1 - \frac{N_2(p_1 - p_1 p_2) \frac{p_1}{p_2}}{N_1(p_2 - p_1 p_2)} \right) \frac{1}{p_1} \right) \\ & = \frac{\Delta n_1 - 1}{p_1 + p_2 - p_1 p_2} + \frac{(\Delta n_1 - 1)(p_2 - p_1 p_2)}{(p_1 + p_2 - p_1 p_2) p_1} \\ & \quad + \frac{\Delta n_2 - 1}{p_1 + p_2 - p_1 p_2} + \frac{(\Delta n_2 - 1)(p_1 - p_1 p_2)}{(p_1 + p_2 - p_1 p_2) p_2} \\ & \quad - (\Delta n_1 - 1) \frac{N_2(p_1 - p_1 p_2) \frac{1}{p_2}}{N_1(p_1 + p_2 - p_1 p_2)} \\ & = \frac{\Delta n_1 - 1}{p_1} + \frac{\Delta n_2 - 1}{p_2} - (\Delta n_1 - 1) \frac{N_2(p_1 - p_1 p_2) \frac{1}{p_2}}{N_1(p_1 + p_2 - p_1 p_2)} \\ & \geq \frac{\Delta n_1 - 1}{p_1} + \left( \frac{\Delta n_1 \lambda_1 - \lambda_1 - \lambda_2}{\lambda_2} - 1 \right) \frac{1}{p_2} \\ & \quad - (\Delta n_1 - 1) \frac{(p_1 - p_1 p_2) \frac{1}{p_2} \lambda_1}{p_1 + p_2 - p_1 p_2 \lambda_2} \quad (19) \\ & = (\Delta n_1 - 1) \left( \frac{1}{p_1} + \frac{\lambda_1}{\lambda_2} \frac{1}{p_2} - \frac{\lambda_1}{\lambda_2} \frac{1}{p_2} \frac{p_1 - p_1 p_2}{p_1 + p_2 - p_1 p_2} \right) - \frac{2}{p_2} \\ & = (\Delta n_1 - 1) \left( \frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2} \right) \lambda_1 - \frac{2}{p_2} \\ & = \left[ \frac{(t_2 - t_1)}{\left( \frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2} \right) \lambda_1} \right] \left( \frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2} \right) \lambda_1 \\ & \quad - \frac{2}{p_2} \\ & \approx t_2 - t_1, \end{aligned} \quad (20)$$

where the inequality in (19) is due to the fact that the difference between  $\Delta n_1 \lambda_1$  and  $\Delta n_2 \lambda_2$  is at most  $\lambda_1 + \lambda_2$ .

Since for any time slot in  $[t_1 + 1, t_3]$  we either send an uncoded or a coded transmission, we must have  $t_3 - t_1 = \text{UT}_1 + \text{UT}_2 + \text{ST}_1 + \text{TCT}$ . Then by (9), (14), and (20), the definition of  $\Delta n_1$  and  $\Delta n_2$ , we have thus 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  $\Delta n_1$  and  $\Delta n_2$  are sufficiently large), we have

$$\mathbb{P}((t_3 - t_1) > (t_2 - t_1)(1 - \epsilon_8) | \mathcal{A}_{t_1}) > 1 - \delta_8. \quad (21)$$

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  $t_1 + (t_2 - t_1)(1 - \epsilon_8)$ . Therefore, at the end of time  $t_1 + (t_2 - t_1)(1 - \epsilon_8)$ , either  $n_1(t)$  must be no larger than  $n_1(t_1) + \Delta n_1$  or  $n_2(t)$  must be no larger than  $n_2(t_1) + \Delta n_2$  with close-to-one probability since we have not reached  $t_3$  yet. (21) thus implies

$$\begin{aligned} & \mathbb{P}(n_2(t_1 + (t_2 - t_1)(1 - \epsilon_8)) \geq n_2(t_1) + \Delta n_2 \quad \& \\ & \quad n_1(t_1 + (t_2 - t_1)(1 - \epsilon_8)) \geq n_1(t_1) + \Delta n_1 | \mathcal{A}_{t_1}) < \delta_8. \end{aligned} \quad (22)$$

By (22) we have

$$\begin{aligned} & \mathbb{P}(n_2(t_1 + (t_2 - t_1)(1 - \epsilon_8)) \leq n_2(t_1) + \Delta n_2 \quad \text{or} \\ & \quad n_1(t_1 + (t_2 - t_1)(1 - \epsilon_8)) \leq n_1(t_1) + \Delta n_1 | \mathcal{A}_{t_1}) > 1 - \delta_8. \end{aligned} \quad (23)$$

Since our algorithm tries to minimize the difference between  $\lambda_1 n_1(t)$  and  $\lambda_2 n_2(t)$ , if  $n_2(t_1 + (t_2 - t_1)(1 - \epsilon_8)) \leq n_2(t_1) + \Delta n_2$ , by the relationship  $\lambda_2 \Delta n_2 \leq \lambda_1 \Delta n_1 + \lambda_1$ , it implies that

$$\begin{aligned} & n_1(t_1 + (t_2 - t_1)(1 - \epsilon_8)) \\ & \leq \frac{\lambda_2 n_2(t_1 + (t_2 - t_1)(1 - \epsilon_8))}{\lambda_1} + 1 \end{aligned} \quad (24)$$

$$\begin{aligned} & \leq \frac{\lambda_2 n_2(t_1) + \lambda_1 \Delta n_1 + \lambda_1}{\lambda_1} + 1 \\ & \leq \frac{\lambda_1 n_1(t_1) + \lambda_2 + \lambda_1 \Delta n_1 + \lambda_1}{\lambda_1} + 1 \end{aligned} \quad (25)$$

$$= n_1(t_1) + \Delta n_1 + \frac{\lambda_2}{\lambda_1} + 2, \quad (26)$$

where (24) due to the fact that  $\lambda_1 n_1(t) \leq \lambda_2 n_2(t) + \lambda_1$  and (25)  $\lambda_2 n_2(t) \leq \lambda_1 n_1(t) + \lambda_2$ .

By the same approach, we have if  $n_1(t_1 + (t_2 - t_1)(1 - \epsilon_8)) \leq n_1(t_1) + \Delta n_1$ , it implies that

$$n_2(t_1 + (t_2 - t_1)(1 - \epsilon_8)) \leq n_2(t_1) + \Delta n_2 + \frac{\lambda_1}{\lambda_2} + 2. \quad (27)$$

So combining (23), (26), and (27), we have

$$\begin{aligned} & \mathbb{P} \left( n_2(t_1 + (t_2 - t_1)(1 - \epsilon_8)) \leq n_2(t_1) + \Delta n_2 + \frac{\lambda_1}{\lambda_2} + 2 \quad \& \\ & \quad n_1(t_1 + (t_2 - t_1)(1 - \epsilon_8)) \leq n_1(t_1) + \Delta n_1 + \frac{\lambda_2}{\lambda_1} + 2 \right. \\ & \quad \left. | \mathcal{A}_{t_1} \right) > 1 - \delta_5. \end{aligned} \quad (28)$$

Then we have

$$\begin{aligned} & \mathbb{P}(n_2(t_1 + (t_2 - t_1)(1 - \epsilon_8)) > n_2(t_1) + \Delta n_2 + \frac{\lambda_1}{\lambda_2} + 2 \\ & \quad | \mathcal{A}_{t_1}) < \delta_5, \end{aligned} \quad (29)$$

and

$$\begin{aligned} \mathbb{P}(n_1(t_1 + (t_2 - t_1)(1 - \epsilon_8)) > n_1(t_1) + \Delta n_1 + \frac{\lambda_2}{\lambda_1} + 2 \\ \left| \mathcal{A}_{t_1} \right\rangle < \delta_8. \end{aligned} \quad (30)$$

That is

$$\begin{aligned} \mathbb{P}(n_2(t_1 + (t_2 - t_1)(1 - \epsilon_8)) \leq n_2(t_1) + \Delta n_2 + \frac{\lambda_1}{\lambda_2} + 2 \\ \left| \mathcal{A}_{t_1} \right\rangle > 1 - \delta_8, \end{aligned} \quad (31)$$

and

$$\begin{aligned} \mathbb{P}(n_1(t_1 + (t_2 - t_1)(1 - \epsilon_8)) \leq n_1(t_1) + \Delta n_1 + \frac{\lambda_2}{\lambda_1} + 2 \\ \left| \mathcal{A}_{t_1} \right\rangle > 1 - \delta_8. \end{aligned} \quad (32)$$

We then notice the following two facts: (i) the difference between  $t_2$  and  $(t_1 + (t_2 - t_1)(1 - \epsilon_8))$  is  $(t_2 - t_1)\epsilon_8$ ; and (ii) for any  $t'_1 < t'_2$ , either  $n_1(t'_2) - n_1(t'_1)$  or  $n_2(t'_2) - n_2(t'_1)$  is no larger than  $t'_2 - t'_1$  since the register  $n_1(t)$  or  $n_2(t)$  at most increments by one in every time slots. As a result, (32) implies

$$\begin{aligned} \mathbb{P}\left(n_1(t_2) - n_1(t_1) \leq \Delta n_1 + \frac{\lambda_2}{\lambda_1} + 2 + (t_2 - t_1)\epsilon_8 \left| \mathcal{A}_{t_1} \right\rangle \right. \\ \left. > 1 - \delta_8. \right. \end{aligned} \quad (33)$$

We can then reuse the above fact (ii) to upper bound the expectation of  $n_1(t_2) - n_1(t_1)$ :

$$\begin{aligned} \mathbb{E}\left\{n_1(t_2) - n_1(t_1) \left| \mathcal{A}_{t_1} \right\rangle\right\} \leq \left(\Delta n_1 + \frac{\lambda_2}{\lambda_1} + 2 + (t_2 - t_1)\epsilon_8\right) \\ \times (1 - \delta_8) + \delta_8(t_2 - t_1). \end{aligned} \quad (34)$$

By noticing that  $\Delta n_1$  is linearly proportional to  $(t_2 - t_1)$  while all other terms are sub-linear (with either a  $\epsilon$  or a  $\delta$  coefficient), (34) thus implies that for any  $\epsilon > 0$ , there exists a sufficiently large  $B$  such that if  $t_2 - t_1 > B$ , then

$$\mathbb{E}\left\{n_1(t_2) - n_1(t_1) \left| \mathcal{A}_{t_1} \right\rangle\right\} \leq \frac{(t_2 - t_1)\gamma(1 + \epsilon)}{\lambda_1}. \quad (35)$$

By similar argument, we have

$$\mathbb{E}\left\{n_2(t_2) - n_2(t_1) \left| \mathcal{A}_{t_1} \right\rangle\right\} \leq \frac{(t_2 - t_1)\gamma(1 + \epsilon)}{r\lambda_2}. \quad (36)$$

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  $H_{1,n}$ ,  $H_{2,n}$ ,  $ST_{1,n}$ ,  $CT_{1,n}$ ,  $CT_{2,n}$  as in (7), (10), and (12), respectively. Note that now these five random variables are no longer independently distributed as the results of one, say  $H_{1,n}$ , may affect the other, say  $CT_{2,n}$ , due to expiration. Define a set of *shadow random variables*  $\tilde{H}_{1,n}$ ,  $\tilde{H}_{2,n}$ ,  $\tilde{ST}_{1,n}$ ,  $\tilde{CT}_{1,n}$ ,  $\tilde{CT}_{2,n}$  that characterize the behaviors when there is no expiration 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. We choose  $\tilde{ST}_{1,n} = ST_{1,n}$  if  $ST_{1,n}$  stops growing due to the single transmission involving  $X_{1,n}$  being received by  $d_1$ . Similarly, we choose  $\tilde{CT}_{1,n} = CT_{1,n}$  if  $CT_{1,n}$  stops growing due to the mixed coded transmission involving  $X_{1,n}$  being received by  $d_1$ . If  $CT_{1,n}$  stops growing due to the expiration of  $X_{1,n}$ , then we let  $\tilde{CT}_{1,n}$  continue to grow as an independent geometric random variable. In this way,  $\tilde{CT}_{1,n}$  mimics the behavior of a system with no expiration and  $\tilde{CT}_{1,n}$  is independent from all other random variables.

Then we need to prove the following version of (21): 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

$$\begin{aligned} \delta_8 &\geq \mathbb{P}(\text{UT}_1 + \text{UT}_2 + \text{ST}_1 + \text{TCT} \leq (t_2 - t_1)(1 - \epsilon_8) \left| \mathcal{A}_{t_1} \right\rangle) \\ &= \mathbb{P}\left(\sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} H_{1,i} + \sum_{j=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2-1} H_{2,j} \right. \\ &\quad \left. + \sum_{j=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} \text{ST}_{1,j} \right. \\ &\quad \left. + \min\left(\sum_{k=1}^{n_1(t_1)+\Delta n_1-1} \text{CT}_{1,k}, \sum_{l=1}^{n_2(t_1)+\Delta n_2-1} \text{CT}_{2,l}\right) \right. \\ &\quad \left. \leq (t_2 - t_1)(1 - \epsilon_8) \left| \mathcal{A}_{t_1} \right\rangle\right) \end{aligned} \quad (37)$$

Note that conditioning on  $\mathcal{A}_{t_1}$ , 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  $H_{1,i}$ ,  $H_{2,j}$ ,  $ST_{1,k}$ ,  $CT_{1,k}$ , and  $CT_{2,l}$  in (37) 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

$$\begin{aligned} &\mathbb{P}\left(\sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} H_{1,i} + \sum_{j=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2-1} H_{2,j} \right. \\ &\quad \left. + \min\left(\sum_{k=1}^{n_1(t_1)+\Delta n_1-1} \text{CT}_{1,k}, \sum_{l=1}^{n_2(t_1)+\Delta n_2-1} \text{CT}_{2,l}\right) \right. \\ &\quad \left. + \sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} \text{ST}_{1,i} \leq (t_2 - t_1)(1 - \epsilon_8) \left| \mathcal{A}_{t_1} \right\rangle\right) \\ &\leq \mathbb{P}\left(\sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} \tilde{H}_{1,i} + \sum_{j=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2-1} \tilde{H}_{2,j} \right. \\ &\quad \left. + \min\left(\sum_{k=n_1(t_1)+1}^{n_1(t_1)+\Delta n_2-1} \tilde{CT}_{1,k}, \sum_{l=n_2(t_1)+1}^{n_2(t_1)+\Delta n_2-1} \tilde{CT}_{2,l}\right) \right. \\ &\quad \left. + \sum_{i=n_1(t_1)+1}^{n_1(t_1)+\Delta n_1-1} \tilde{ST}_{1,i} \leq (t_2 - t_1)(1 - \epsilon_8) \left| \mathcal{A}_{t_1} \right\rangle\right) \end{aligned} \quad (38)$$

since for those realizations, the shadow random variables and the actual random variables of packets with indices  $\geq n_1(t_1)$  for session 1 and packets with indices  $\geq n_2(t_1)$  for session 2, and transmitted between  $[t_1+1, t_2]$ , have the same probability. Since (21) holds for the case without expiration, (38) can thus be made smaller than  $\delta_8$  with sufficiently large  $B_8$ . (37) is thus proven. We can then follow the same analysis as in (21) to (35).

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,  $\frac{\lambda_1 p_1 (p_1 - p_1 p_2)}{\lambda_2 p_2 (p_2 - p_1 p_2)} > 1$ , and  $\frac{1}{\gamma} = \frac{1/\lambda_1}{p_1 + p_2 - p_1 p_2} + \frac{1/\lambda_2}{p_2}$ ). Then the proof of Lemma 2 is complete. ■

For the following, we would first present the proof for Proposition 2 of the over-provisioned case ( $\gamma \geq 1$ ).

*Proof:* For ease of exposition, we first assume that user 1 is the leading user. Since we are considering the over-provisioned case, we have  $0 < \frac{1}{\gamma} = \frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2} \leq 1$ . We define  $q_j(t) \triangleq n_j(t) - \frac{\gamma t}{\lambda_j(1-\epsilon')}$  for  $j = 1, 2$ , where  $\epsilon' > 0$  is a small number. Suppose  $t_2 = t_1 + B_0$ , where  $B_0$  will be chosen shortly. Note that by the definition of  $q_1(t)$  and  $q_2(t)$  and by Lemma 1, we have  $q_2(t) \geq \frac{n_1(t)\lambda_1 - \lambda_1}{\lambda_2} - \frac{\gamma t}{\lambda_2(1-\epsilon')} = \frac{\lambda_1}{\lambda_2} q_1(t) - \frac{\lambda_1}{\lambda_2}$ . One can also check that three conditions  $q_1(t_1) > \frac{B_0}{\lambda_1} + 1$  and  $q_2(t_1) > \frac{B_0}{\lambda_2}$ ,  $t_2 = t_1 + B_0$  imply the condition that  $t_2 < \min(\lambda_1 n_1(t_1), \lambda_2 n_2(t_1))$  in Lemma 2. As a result, we can prove that for any  $\epsilon > 0$ , there exists a  $B > 0$  such that for any  $B_0 > B$

$$\begin{aligned} & \mathbb{E}\left\{q_1(t_1 + B_0) - q_1(t_1) \mid q_1(t_1) > \frac{B_0}{\lambda_1} + 1\right\} \\ &= \mathbb{E}\left\{q_1(t_1 + B_0) - q_1(t_1) \mid q_1(t_1) > \frac{B_0}{\lambda_1} + 1, q_2(t_1) > \frac{B_0}{\lambda_2}\right\} \\ &= \mathbb{E}\left\{n_1(t_1 + B_0) - n_1(t_1) \mid q_1(t_1) > \frac{B_0}{\lambda_1} + 1, q_2(t_1) > \frac{B_0}{\lambda_2}\right\} \\ &\quad - \frac{\gamma B_0}{\lambda_1(1-\epsilon')} \\ &\leq \frac{\gamma B_0(1+\epsilon)}{\lambda_1} - \frac{r B_0}{\lambda_1(1-\epsilon')} < 0, \end{aligned} \quad (39)$$

where the strict inequality in the second step of (39) is established by choosing a sufficiently small  $\epsilon > 0$ . Similarly

$$\mathbb{E}\left\{q_2(t_1 + B_0) - q_2(t_1) \mid q_2(t_1) > \frac{B_0}{\lambda_2} + 1\right\} < 0. \quad (40)$$

Eq. (39) and (40) show that both  $q_1(t)$  and  $q_2(t)$  have negative drift. Since  $q_1(t)$  has a negative drift, it implies that for any  $\epsilon_1, \epsilon' > 0$ , there exists a  $t_0 > 0$  such that  $\mathbb{P}(q_1(t) < \epsilon' t) > 1 - \epsilon_1$ , for all  $t > t_0$ . Then the following

inequality holds for any  $t > t_0$ ,

$$\begin{aligned} \mathbb{E}\{n_1(t)\} &= \mathbb{E}\left\{\frac{\gamma t}{\lambda_1(1-\epsilon')} + q_1(t)\right\} \\ &= \mathbb{E}\left\{\frac{\gamma t}{\lambda_1(1-\epsilon')} + q_1(t) \mid q_1(t) < \epsilon' t\right\} \mathbb{P}(q_1(t) < \epsilon' t) \\ &\quad + \mathbb{E}\{n_1(t) \mid q_1(t) \geq \epsilon' t\} \mathbb{P}(q_1(t) \geq \epsilon' t) \\ &\leq \left(\frac{\gamma t}{\lambda_1(1-\epsilon')} + \epsilon' t\right)(1 - \epsilon_1) + t\epsilon_1, \end{aligned} \quad (41)$$

where (42) is because  $n_1(t)$  is always upper bounded by  $t$  regardless whether  $q_1(t) \geq \epsilon' t$  or not. Eq. (42) shows that the expectation  $\mathbb{E}\{n_1(t)\}$  is upper bounded by  $\frac{\gamma t}{\lambda_1} + o(t)$ . Similarly, we have  $\mathbb{E}\{n_2(t)\} \leq \frac{\gamma t}{\lambda_2} + o(t)$ . We define  $T_j(t)$  as the number of time slots when the BS transmits an uncoded packet for session  $j$  up to time  $t$ . 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

$$\mathbb{E}\{T_2(t)\} \leq \mathbb{E}\{n_2(t)\} \frac{1}{p_1 + p_2 - p_1 p_2}, \quad (43)$$

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 retransmitted until user 1 receives it even after it has been received by user 2.  $T_1(t)$  is comprised of two types of transmissions: The first type is when the BS transmits uncoded packets of session 1. The other type is when the BS transmit session-1 packets that have status being **uncoded-Tx-only** and have been received by user 2 first (in which case the BS continues to transmit this type of packets until user 1 receives it). The first part can be upper bounded by  $\mathbb{E}\{n_1(t)\} \frac{1}{p_1 + p_2 - p_1 p_2}$ . We use  $\text{UCO}(t)$  to denote the total number of time slots that are used to “retransmit” some coded opportunities of user 1 whose status have been set to **uncoded-Tx-only** during the interval  $[1, t]$  (the second part of  $T_1(t)$ ). By the same argument as used in the proof for Lemma 2, we thus have

$$\begin{aligned} \mathbb{E}\{\text{UCO}(t)\} &\leq \mathbb{E}\{n_1(t)\} \left(1 - \frac{N_2(p_1 - p_1 p_2) \frac{p_1}{p_2}}{N_1(p_2 - p_1 p_2)}\right) \\ &\quad \times \left(\frac{1}{p_1} - \frac{1}{p_1 + p_2 - p_1 p_2}\right), \end{aligned} \quad (44)$$

where 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

$$\mathbb{E}\{T_1(t)\} \leq \mathbb{E}\{n_1(t)\} \frac{1}{p_1} \left(1 - \frac{\lambda_1 p_1 (p_1 - p_1 p_2)}{\lambda_2 p_2 (p_1 + p_2 - p_1 p_2)}\right). \quad (45)$$

Note that when we transmit an uncoded packet for session 1, the expected “reward” is  $p_1$  since only user 1 can get benefits from this transmission. When we transmit a coded packet, the expected reward for user 1 is  $p_1$  and reward for user 2 is  $p_2$  since both destinations can benefit from the

coded transmission. As a result, for sufficiently large  $t$ , the expected total rewards for user 1 is lower bounded by

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

where the inequality follows from  $\mathbb{E}\{n_2(t)\} \leq \frac{rt}{\lambda_2} + o(t)$  and (43). When  $t = \lambda_1 N_1 / \gamma$ , we have  $\mathbb{E}\{N_1^{\text{success}}\} = N_1 - o(t)$ . As a result, the achievable rate  $\frac{N_1^{\text{success}}}{\lambda_1 N_1}$  approaches  $\frac{1}{\lambda_1}$  for sufficiently large  $N_1$ . Similarly, the expected total rewards for user 2 is lower bounded by

$$\begin{aligned} & \mathbb{E}\{N_2^{\text{success}}\} \tag{46} \\ &= p_2 \mathbb{E}\{T_2(t)\} + p_2 \mathbb{E}\{t - T_1(t) - T_2(t)\} \tag{47} \\ &= p_2 t - p_2 \mathbb{E}\{T_1(t)\} \tag{48} \\ &\geq p_2 t - p_2 \frac{\gamma t}{\lambda_1} \left( \frac{1}{p_1} - \frac{N_2(p_1 - p_1 p_2) \frac{1}{p_2}}{(p_1 + p_2 - p_1 p_2) N_1} \right) - o(t) \\ &= \frac{\gamma t}{\lambda_2} - o(t). \tag{49} \end{aligned}$$

When  $t = \lambda_2 N_2 / \gamma$ , we have  $\mathbb{E}\{N_2^{\text{success}}\} = N_2 - o(t)$ . Hence, the achievable rate  $\frac{N_2^{\text{success}}}{\lambda_2 N_2}$  also approaches  $\frac{1}{\lambda_2}$  for sufficiently large  $N_2$ . Proposition 2 is thus proved for the case  $\gamma \geq 1$  and user 1 being the leading user. Symmetrically, we can prove Proposition 2 when user 2 is the leading user for the over-provisioned case. ■

The under-provisioned case is similar and we would provide the sketch in the following. To prove the under-provisioned case for Proposition 2, we first need to prove the under-provisioned case for Lemma 2. That is, we need 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  $j = 1, 2$ ,

$$\begin{aligned} & \mathbb{E}\left\{n_j(t_2) - n_j(t_1) \mid t_2 < \min(\lambda_1 n_1(t_1), \lambda_2 n_2(t_1))\right\} \\ &\leq \frac{(t_2 - t_1)(1 + \epsilon)}{\lambda_j}. \end{aligned} \tag{50}$$

Note that for under-provisioned case, we have  $\gamma < 1$ .

*Proof:*

$$\begin{aligned} \text{Define } \Delta n_1 &= \left\lceil \frac{(t_2 - t_1)}{\left(\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}\right) \lambda_1} \right\rceil + 1, \\ \Delta n_2 &= \left\lceil \frac{(t_2 - t_1)}{\left(\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}\right) \lambda_2} \right\rceil + 1. \end{aligned}$$

We use  $t_3$  to denote the (random) time slot for which in the end of time  $t_3$ , the BS has scheduled real transmission of  $\Delta n_1$  uncoded packets for session 1 and  $\Delta n_2$  uncoded packets for session 2, respectively. We then relabel the next  $\Delta n_1$  packets including packet  $n_1(t_1)$  (that have been

transmitted by the BS) from session 1, as  $\bar{n}_1(t_1), \dots, \bar{n}_1(t_1) + \Delta n_1 - 1$ . We also relabel the next  $\Delta n_2$  packets (that have been transmitted by the BS) including packet  $n_2(t_1)$  from session 2, as  $\bar{n}_2(t_1), \dots, \bar{n}_2(t_1) + \Delta n_2 - 1$ .

We first examine how long it takes before the BS finishes transmitting packets  $\bar{n}_1(t), \dots, \bar{n}_1(t_1) + \Delta n_1 - 1$ , and the the BS finishes transmitting packets  $\bar{n}_2(t), \dots, \bar{n}_2(t_1) + \Delta n_2 - 1$ . That is, we want to understand the time  $t_3$ . We then would examine at the end of time  $t_3$ , how would  $n_1(t_3)$  and  $n_2(t_3)$  be. That is, we want to investigate how many packets for each session that have been transmitted by the BS or discarded without transmission due to congestion control by the BS.

By our congestion control algorithm, whenever  $x_1$  is increased by 1, then for session 1 the BS would schedule to transmit the next uncoded packet. Recall that in the under-provisioned case,  $\gamma < 1$ , that is  $\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2} > 1$ . So after the BS finishes with transmitting  $\Delta n_1$  uncoded packets from session 1, the register  $n_1$  is at most increased by

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

and the register  $n_2$  is at most increased by

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

After this, we can apply the similar proof as in Lemma 2. We can show that  $t_3$ , with high probability, is no less than  $t_1 + (t_2 - t_1)(1 - \epsilon')$ . Then following the same arguments shown in the proof of Lemma 2, we have

$$\mathbb{E}\left\{n_1(t_2) - n_1(t_1) \mid \mathcal{A}_{t_1}\right\} \leq \frac{(t_2 - t_1)(1 + \epsilon)}{\lambda_1}. \tag{51}$$

By similar argument, we have

$$\mathbb{E}\left\{n_2(t_2) - n_2(t_1) \mid \mathcal{A}_{t_1}\right\} \leq \frac{(t_2 - t_1)(1 + \epsilon)}{\lambda_2}. \tag{52}$$

Now we have shown the proof for the under-provisioned case of Lemma 2. Next we are going to show the maximum throughput that can be achieved by our scheme. Still, for ease of exposition, we assume that user 1 is the leading user. ■

We now define  $q_1(t) \triangleq n_1(t) - \frac{t}{\lambda_1(1-\epsilon')}$  and  $q_2(t) \triangleq n_2(t) - \frac{t}{\lambda_2(1-\epsilon')}$ . By the results in (51) and (52), we can then show the negative drift of  $q_1(t)$  and  $q_2(t)$  by similar approaches shown in (39) and (40). When choosing  $q_1(t_1) > \frac{B_0}{\lambda_1} + 1$  and  $q_2(t_1) > \frac{B_0}{\lambda_2}$ ,  $t_2 = t_1 + B_0$ , we can satisfy the conditions that  $t_2 < \lambda_1 n_1(t_1), t_2 < \lambda_2 n_2(t_1)$  in Lemma 2. More specifically, for any  $\epsilon > 0$ , there exists a  $B > 0$  such

that for any  $B_0 > B$

$$\begin{aligned}
& \mathbb{E}\left\{q_1(t_1 + B_0) - q_1(t_1) \middle| q_1(t_1) > \frac{B_0}{\lambda_1} + 1\right\} \\
&= \mathbb{E}\left\{q_1(t_1 + B_0) - q_1(t_1) \middle| q_1(t_1) > \frac{B_0}{\lambda_1} + 1, q_2(t_1) > \frac{B_0}{\lambda_2}\right\} \\
&= \mathbb{E}\left\{n_1(t_1 + B_0) - n_1(t_1) \middle| q_1(t_1) > \frac{B_0}{\lambda_1} + 1, q_2(t_1) > \frac{B_0}{\lambda_2}\right\} \\
&\quad - \frac{B_0}{\lambda_1} \\
&\leq \frac{(t_2 - t_1)(1 + \epsilon)}{\lambda_1} - \frac{B_0}{\lambda_1(1 - \epsilon')} < 0, \tag{53}
\end{aligned}$$

where the negativeness is established by choosing a sufficiently small  $\epsilon > 0$ . Similarly

$$\mathbb{E}\left\{q_2(t_1 + B_0) - q_2(t_1) \middle| q_2(t_1) > \frac{B_0}{\lambda_2} + 1\right\} < 0. \tag{54}$$

Since  $q_1(t)$  has a negative drift, it implies that for any  $\epsilon_1, \epsilon' > 0$ , there exists a  $t_0 > 0$  such that  $\mathbb{P}(q_1(t) < \epsilon't) > 1 - \epsilon_1$ , for all  $t > t_0$ .

Using the negative drift of  $q_1(t)$ , we have for any  $t > t_0$ ,

$$\mathbb{E}\{n_1(t)\} \leq \frac{t}{\lambda_1} + o(t), \tag{55}$$

where  $o(t)$  is a sublinear term, by similar steps shown in (42). Meanwhile, we also have

$$\mathbb{E}\{n_2(t)\} \leq \frac{t}{\lambda_2} + o(t). \tag{56}$$

We still use  $T_1(t)$  to denote the number of time slots when the BS transmits an uncoded packet for session 1 up to time  $t$ ;  $T_2(t)$  as the number of time slots when the BS transmits an uncoded packet for session 2 up to time  $t$ . By our submodule FNISS, for under provisioned case, the BS would only choose to transmit at most  $n_2(t) \left( \frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}} \right)$  out of the first  $n_2(t)$  packets in session 2. Since for every uncoded packet from session 2 that has been chosen to transmit, the BS will transmit it until it has been received by at least one user, we have

$$\begin{aligned}
& \mathbb{E}\{T_2(t)\} \\
&\leq \mathbb{E}\{n_2(t)\} \left( \frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}} \right) \frac{1}{p_1 + p_2 - p_1 p_2}, \tag{57}
\end{aligned}$$

where the inequality is because some uncoded packets are expired before they can be received by any user, so the expected transmission time for each packet is shortened.

For session 1, since user 1 is the leading user, some packets would be retransmitted until user 1 receives it, if it has been received by user 2 first.  $T_1(t)$  is 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. The first part can be upper bounded by  $\mathbb{E}\{n_1(t)\} \left( \frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}} \right) \frac{1}{p_1 + p_2 - p_1 p_2}$  (some packets need to be dropped by the congestion control mechanism in FNISS). For the second part, to illustrate the

calculation, we use  $RN(t)$  as number of coding opportunities of user 1 that the BS decides to retransmit until user 1 receives, during the interval  $[1, t]$ .

We use  $UCO(t)$  to denote the total number of time slots that are used to ‘‘retransmit’’ some coded opportunities of user 1 whose status have been set to **uncoded-Tx-only** during the interval  $[1, t]$  (the second part of  $T_1(t)$ ). By the same argument as used in the proof for Lemma 2, we thus have

$$\begin{aligned}
& \mathbb{E}\{UCO(t)\} \frac{1}{p_1} = \frac{1}{p_1} \left( 1 - \frac{N_2(p_1 - p_1 p_2) \frac{p_1}{p_2}}{N_1(p_2 - p_1 p_2)} \right) \\
&\quad \times \left( \frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}} \right) \frac{p_2 - p_1 p_2}{p_1 + p_2 - p_1 p_2} \mathbb{E}\{n_1(t)\}. \tag{58}
\end{aligned}$$

Combining the first and second part, we obtain

$$\mathbb{E}\{T_1(t)\} \leq \mathbb{E}\{n_1(t)\} \left( \frac{1}{p_1 + p_2 - p_1 p_2} + \frac{p_2 - p_1 p_2}{p_1 + p_2 - p_1 p_2} \frac{1}{p_1} \right) \tag{59}$$

$$\begin{aligned}
& - \frac{p_2 - p_1 p_2}{p_1 + p_2 - p_1 p_2} \frac{N_2(p_1 - p_1 p_2) \frac{p_1}{p_2}}{N_1(p_2 - p_1 p_2)} \frac{1}{p_1} \\
&\quad \times \left( \frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}} \right) \\
&= \mathbb{E}\{n_1(t)\} \left( \frac{1}{p_1} - \frac{N_2(p_1 - p_1 p_2) \frac{1}{p_2}}{(p_1 + p_2 - p_1 p_2) N_1} \right) \\
&\quad \times \left( \frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}} \right), \tag{60}
\end{aligned}$$

where the inequality is because some uncoded packets are expired before they can be received by any user, so the expected transmission time for each packet is shortened; or because some coding opportunities may expire before they can be received through coded transmission.

Note that when we transmit an uncoded packet for session 1, the expected ‘‘reward’’ is  $p_1$  since only user 1 can get benefits from this transmission. When we transmit a coded packet, the expected reward for user 1 is  $p_1$  and reward for user 2 is  $p_2$  since both destinations can benefit. As a result, for sufficiently large  $t$ , the expected total rewards for user 1 is lower bounded by

$$\mathbb{E}\{N_1^{\text{success}}\} \tag{62}$$

$$= p_1 \mathbb{E}\{T_1(t)\} + p_1 \mathbb{E}\{t - T_1(t) - T_2(t)\} \tag{63}$$

$$= p_1 t - p_1 \mathbb{E}\{T_2(t)\} \tag{64}$$

$$\geq p_1 t - p_1 \frac{t}{\lambda_2} \frac{1}{p_1 + p_2 - p_1 p_2} \left( \frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}} \right)$$

$$= \frac{t}{\lambda_1} \left( \frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}} \right). \tag{65}$$

When  $t$  is  $\lambda_1 N_1$ , we have  $\mathbb{E}\{N_1^{\text{success}}\} = N_1 \left( \frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}} \right)$ . The expected total rewards for

user 2 is lower bounded by

$$\mathbb{E}\{N_2^{\text{success}}\} \quad (66)$$

$$= p_2 \mathbb{E}\{T_2(t)\} + p_2 \mathbb{E}\{t - T_1(t) - T_2(t)\} \quad (67)$$

$$= p_2 t - p_2 \mathbb{E}\{T_1(t)\} \quad (68)$$

$$\begin{aligned} &\geq p_2 t - p_2 \frac{t}{\lambda_2} \left( \frac{1}{p_1} - \frac{N_2(p_1 - p_1 p_2) \frac{1}{p_2}}{(p_1 + p_2 - p_1 p_2) N_1} \right) \\ &\quad \times \left( \frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}} \right) \\ &= \frac{t}{\lambda_2} \left( \frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}} \right). \end{aligned} \quad (69)$$

When  $t$  is  $\lambda_2 N_2$ , we have  $\mathbb{E}\{N_2^{\text{success}}\} = N_2 \left( \frac{1}{\frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}} \right)$ . We have shown the throughput performance for the under-provisioned case when user 1 is a leading user. We can apply the same approaches for the scenario when user 2 is a leading user. Recall that  $\gamma$  denotes the value  $\min \left( \frac{1/\lambda_1}{p_1} + \frac{1/\lambda_2}{p_1 + p_2 - p_1 p_2}, \frac{1/\lambda_1}{p_1 + p_2 - p_1 p_2} + \frac{1/\lambda_2}{p_2} \right)$ . Thus we can conclude that, for under-provisioned case, the throughput for user 1 is  $N_1 \gamma$ , and the throughput for user 1 is  $N_2 \gamma$ .

## V. SIMULATION

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

### 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  $(\lambda_1, \lambda_2)$  being (2,4), (3,4), (4,4), (5,4), and (6,4), respectively (we name them as case 1 to case 5, respectively). For all cases we use  $N_1 = 40000$ . Recall that we require  $\lambda_1 N_1 = \lambda_2 N_2$ , and 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. 1, i.e., according to (1) and (2), as shown by the area beneath the two solid lines. We then use different markers to denote the normalized throughput  $\left( \frac{N_1^{\text{success}}}{\lambda_1 N_1}, \frac{N_2^{\text{success}}}{\lambda_2 N_2} \right)$  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. Note that case 1 represents the under-provisioned setting, while the other cases represents the over-provisioned setting. We observe that in all cases, the achievable throughput is very close to the upper bound.

### B. Performance for Small $N_1$ and $N_2$

In Fig. 2 we plot the normalized throughput for both users when  $N_1$  and  $N_2$  are small. We use the same channel parameters  $p_1 = 0.5$  and  $p_2 = 0.6$ , and the same deadlines  $(\lambda_1, \lambda_2)$  being (2,4), (3,4), (4,4), (5,4), and (6,4), respectively. But we use much smaller file sizes:  $(N_1, N_2)$  being (400, 200),

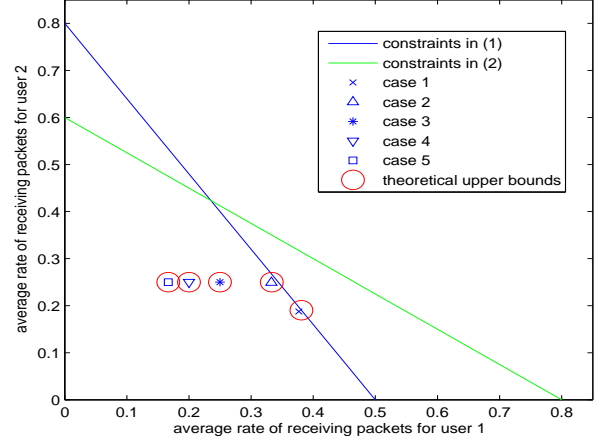


Fig. 1. Average rate of receiving packets for user 1 and user 2 when  $N_1$  and  $N_2$  are large.

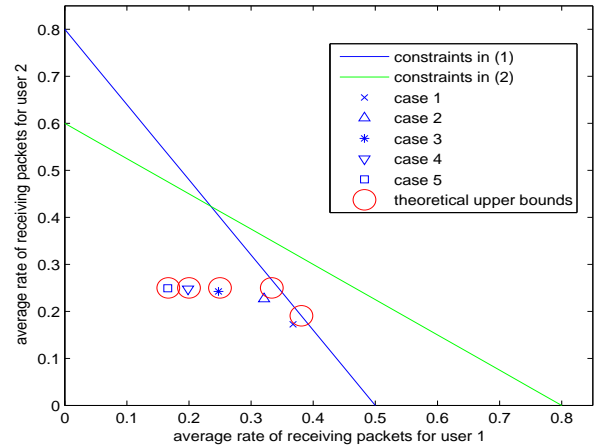


Fig. 2. Average rate of receiving packets for user 1 and user 2 when  $N_1$  and  $N_2$  are small.

(400, 300), (400, 400), (400, 500), and (400, 600). In Fig. 2, the circles still indicate the theoretical upper bound of both sessions; we also plot the achieved normalized throughput for all 5 cases. 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.

## VI. CONCLUSION AND DISCUSSION

In this work, we have studied inter-session network coding for sending two unicast sessions over an unreliable wireless channel. We consider two unicast sessions under heterogeneous channel conditions and heterogeneous deadline constraints. We develop immediately-decodable network coding (IDNC) schemes for controlling packet transmissions for the unicast sessions in order to maximize the normalized throughput subject to hard deadline constraints. Our proposed scheme is not only proved to be asymptotically optimal in the limit of large file sizes, it is also shown in our

simulations to achieve close-to-optimal throughput for small file sizes.

The analysis in this paper assumes that the channel statistics are known to the BS, and perfect feedback is provided to the BS after each time-slot. In practical systems, the assumption of instant, noise-free feedback may no longer hold, and we also might not know the channel before-hand. Our proposed scheme could be adapted to fit these practical settings. For the delayed- and lossy-feedback setting, we can modify our scheme similar to the one in Section V of [6]. On the other hand, if the channel characteristic is not known by the BS, then we can estimate  $p_1$  and  $p_2$  by counting the number of the “ACKs” that the BS has received for each channel. After an initial learning period, the estimate will be close to the real value. We can then use the channel estimate to schedule transmissions. Our future work will analytically quantify the performance of IDNC schemes in such practical settings.

#### 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] 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.
- [8] A. Eryilmaz and D. Lun, “Control for inter-session network coding,” in *NetCod*, 2007.
- [9] 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.
- [10] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, “XORs in the air: Practical wireless network coding,” in *Proc. of ACM SIGCOMM*, 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 in a broadcast packet erasure channel approach,” *IEEE Trans. Inform. Theory*.
- [13] H. Seferoglu and A. Markopoulou, “ $I^2nc$ : 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] J. Barros, R. Costa, D. Munaretto, and J. Widmer, “Effective delay control in online network coding,” in *Proc. of INFOCOM*, 2009.
- [22] D. Nguyen, T. Nguyen, and B. Bose, “Wireless broadcast using network coding,” in *NetCod*, 2007.