This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2010

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

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

***Abstract*—Multimedia streaming applications have stringent QoS requirements. Typically each packet is associated with a packet delivery deadline. This work models and considers real-time streaming broadcast for stored-video over the downlink of a single cell. The broadcast capacity of the system subject to deadline constraints are derived for both uncoded and coded wireless broadcast schemes. Even under the deadline requirements, it is shown in this work that network coding is asymptotically throughput-optimal and can strictly outperform the best non-coding policy by analytically quantifying the optimal capacity when the file size is sufficiently large. A simple network coding policy is also proposed that achieves the asymptotic capacity while maintaining finite transmission delay (queueing + decoding delay). A new temporal-queue-length-based Lyapunov function is used to prove the optimality of this policy. Simulation shows that the simple coding policy outperforms the best non-coding policies even for broadcasting files of small sizes.**

I. Introduction

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

One way of wireless downlink broadcast is to transmit each packet without any coding while bookkeeping the reception status for each individual receiver, respectively. Those packets that are not received successfully for some receivers are retransmitted at a later time. Recently, a new class of network-coding-based broadcast schemes emerges, which performs information mixing among different packets before transmission. When there are no deadline constraints, it is well known that broadcasting coded packets can achieve higher throughput than uncoded policies [1]–[4]. Practical generation-based network coding schemes have been devised in [5], [6], which takes into account asynchronicity and lossy channels in a wireless network. Both the generation-based [5] and the queue-based schemes [6] do not optimize the delay characteristics of the problem. Nonetheless, a commonly referred rule of thumb is that the larger the throughput of the system, the more packets need to be coded together (thus larger generations or longer queues), which in turn causes longer delay as the users need to accumulate a larger number of coded packets before being able to decode a single information packet. Similar to the setting of the 1-hop broadcast session, COPE [7] encodes packets from *different unicast sessions* that enables *1-hop intersession decoding*. The corresponding scheduling and coding policy that decides how to mix packets from different sessions is studied in [8].

In this paper, we are interested in using network coding to improve the throughput for delay-sensitive applications in stored-video streaming settings. For delay-sensitive applications, several network coding works have been proposed based on different delay metrics. For example, the authors of [9]–[11] focus on minimizing the *decoding delay*. In particular, [9] discusses how different methods of encoding can affect the decoding delay in an error-free network from the information-theoretic perspective. The quantification of gains in delay performance of network coding are studied in [10]. [12] focuses on the total transmission delay. Their coding module achieves optimal throughput and is conjectured to simultaneously achieve asymptotically minimal delay. [13] proposes a coding scheme assisted by uncoded transmission that reduces the transmission delay. However, none of the above works directly focuses on the stored-video streaming applications, which prevail in the Internet video watching activities.

In constrast to these prior works, in this work we specifically study the stored-video streaming applications, in which all packets of the entire video file are are stored at the server and available from the beginning. Within this framework, we characterize the optimal capacity regions (the amount of packets that are received before deadlines) for both uncoded and coded solutions and account for the overall transmission delay (i.e., queueing plus coding delays). The setting of this work is mostly related to the following works. [14] considers a similar hard-deadline-constrained, throughput optimization problem for *multiple unicast flows without network coding* and focuses on scheduling/balancing between multiple flows.

In comparison, in this work we characterize the asymptotic capacity of the uncoded transmission by a linear programming formulation, and explicitly quantify the performance of a simple online network coding scheme and prove its asymptotic optimality. This hard deadline constrained problem with network coding was briefly considered in [11] with an order analysis of the queue-length growth rate. The network coding protocol under consideration can be viewed as a generalization of the schemes in [11], [15] for the deadline constrained setting. This work proves the asymptotic optimality of the simple network coding scheme (when the file size is sufficiently large) for the deadline-constrained traffic. Simulation results show that the network coding scheme outperforms the best non-coding scheme even when the file size is small.

The detailed contributions of this work are as follows. We derive the maximum achievable rate of uncoded transmission via a LP formulation, which serves as the baseline when quantifying the deadline-constrained throughput gain of network coding. For the network coded policies, we focus on a *universal* network coding scheme, which does not require the knowledge about the delivery rate p of the wireless broadcast channel. We prove that this scheme achieves the maximum broadcast channel capacity asymptotically for large N in the 2-user case even when all the packets are subject to deadline constraints. We also prove that when N tends to infinity, the total transmission delay (from packet arrival at the base station to decoding at the individual user) remains finite with probability one. These results are in sharp contrast with the existing observations that the throughput improvement of network coding is at the expense of longer delay. Extensive simulations are conducted, which shows that our predicted asymptotic throughput is closely related to the throughput performance even for small finite N. The simulation results also show that this network-coding scheme also outperforms the uncoded schemes for finite small N (packets/file).

II. System Model

We consider the downlink of a single cell in which the base station (BS) broadcasts a file of N packets to several users (for ease of exposition, in this paper, we will mainly focus on the case with 2 users). We assume that time is slotted. Each packet $n = 1, 2, \ldots, N$ has a deadline t_n, after which the packet is no longer useful. To model stored video streaming applications, we assume that the deadlines of the n-th packet are the same for all users, and are of the form

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

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

We consider random and unreliable wireless channels. That is, a packet broadcast from the BS may be received by all users, a subset of users, or no users at all, depending on the random channel conditions. Suppose a packet is transmitted at the t-th time slot. We use $C_j(t) = 1$ to denote that user j can receive the packet successfully, and $C_j(t) = 0$, otherwise. We assume that $\mathsf{P}(C_j(t) = 1) = p$, $j = 1, 2$, and $C_j(t)$ is independently and identically distributed (i.i.d.) across users and time slots. We assume that at the end of each time slot, the BS has perfect feedback whether the packet has been successfully received by each user. Note that based on the feedback, the BS can adapt different transmission strategies and decide whether to send uncoded or coded packet at the next time slot.

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

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

If coding is prohibited, the source can only transmit uncoded packets. Suppose packet n is transmitted at time t, and only one user has received packet n successfully. The BS may decide to retransmit the same packet n for the other user or may decide to move to the next packet $n+1$ to enhance the chance that packet $n+1$ can be received before its deadline. If coding across different packets is allowed, then the BS can code *any* unexpired packets together and broadcast to all users.

Our goal is to design a coding/scheduling policy that maximizes the number of successful (not-expired) packet transmissions. Let $D_j(n) = 1$ if user j can successfully decode/recover packet n from all the coded/uncoded packets received before its deadline λn; and $D_j(n) = 0$, otherwise. We define the total number of successes N_{success} as $\sum_{n=1}^{N} D_1(n) + D_2(n)$. Specifically, we would like to maximize the normalized expected throughput given by $\frac{\mathbb{E}\{N_{\text{success}}\}}{2N}$. We are interested in studying the throughput improvement of network coding versus the best non-coding policy subject to hard deadline constraints.

III. The Optimal Throughput for Uncoded Transmission

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

Fig. 1 illustrates the construction of w_0 to w_2 for a given policy and channel realization. Before all the transmissions, all w_i are set to be 0. At the beginning of $t = 1$, packet

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2010

1 has not been received by any user before, so packet 1 is classified as type-0. Since the BS schedules a type-0 packet for $t = 1$, w_0 is increased by 1. Similarly, packets 2 and 3 scheduled at times 2 and 3 are also of type-0, which contributes to the increment of w_0 at times 2 and 3. At the beginning of $t = 4$ the BS decides to retransmit packet 2 according to the underlying policy. (Here we allow any arbitrary policies including both optimal and suboptimal ones.) Since packet 2 has been received by both users at the end of $t = 2$, at time slot 4 packet 2 is classified as type-2. Therefore w_2 increases. For $t = 5$, packet 1 has been received by 1 user already and is thus classified as type-1. Therefore, w_1 is increased by 1. After all 9 transmissions, we have that $w_0 = 5$, $w_1 = 2$ and $w_2 = 2$. Note that w_j, $j = 0, 1, 2$ are random variables depending on the channel realization and the underlying policy.

Let $\overline{w}_0$, $\overline{w}_1$ and $\overline{w}_2$ denote the expectation of w_0, w_1 and w_2, respectively. When the BS transmits a new packet to both users, the expected reward in the time slot is exactly $2p$. And when the BS transmits a packet that has been received by one user, the expected reward in the time slot is p. Transmitting a packet that has already been received by 2 users would not improve the throughput. Using the optional sampling theorem for Martingales, we can show that the expected total number of successes is $2p\overline{w}_0 + p\overline{w}_1$.

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

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

By noting that $w_0 + w_1 + w_2 = \lambda N$ and by the conservation law of type-0 packets in a similar way as in (1), we have

$$\mathbb{E}\{N_{\text{success}}\} = 2p\overline{w}_0 + p\overline{w}_1 + 0 \cdot \overline{w}_2 \tag{2}$$

$$\begin{aligned} \text{s.t.} \quad & \overline{w}_0 + \overline{w}_1 + \overline{w}_2 \leq \lambda N \\ & \overline{w}_0 (2p - p^2) \leq N \\ & \overline{w}_1 p \leq \overline{w}_0 (2p(1-p)) \\ & \overline{w}_0, \overline{w}_1, \overline{w}_2 \geq 0 \end{aligned} \tag{3}$$

Since the above constraints hold for any policy, we can thus upper bound the best achievable rate for uncoded transmission by maximizing (2) subject to the constraints in (3). A closed-form solution to this linear program then produces the following upper bound (4).

$$\frac{\mathbb{E}\{N_{\text{success}}\}}{2N} \leq \begin{cases} 1 & \text{if } \frac{\lambda+1-\sqrt{\lambda^2+1-\lambda}}{\lambda} < p \leq 1 \\ \frac{1}{2}(\lambda p + \frac{1}{2-p}) & \text{if } \frac{\lambda-\sqrt{\lambda^2-\lambda}}{\lambda} < p \leq \frac{\lambda+1-\sqrt{\lambda^2+1-\lambda}}{\lambda} \\ \lambda p & \text{if } 0 < p \leq \frac{\lambda-\sqrt{\lambda^2-\lambda}}{\lambda} \end{cases}. \tag{4}$$

The detailed derivations are provided in [16]. In [16] we also devise an optimal non-coding scheduling policy based on finite-horizon dynamic programming (DP), for which the optimal expected throughput can be computed numerically. In addition, we provide a proof to show that the proposed upper bound is achievable by a constructive non-coding scheme, so that this bound can indeed be achieved by the optimal non-coding DP scheme, which shows that (4) is indeed the capacity of any non-coding policies.

IV. A Simple And Asymptotically Optimal Policy for Coded Transmission

In this section, we will propose a novel network coding scheme and use a new Lyapunov function to prove that the proposed network coding scheme is asymptotically throughput-optimal when $N \to \infty$. Jointly the results in Sections III and IV quantify the throughput improvement of network coding for deadline-constrained systems.

A. A Simple Network Coding Policy

We first propose a novel transmission scheme that uses network coding. (A similar scheme was proposed in [11] and [15]. However, they did not consider hard deadline constraints, and did not carry out the corresponding capacity analysis.)

Define L_j, $j = 1, 2$, to be the list of unexpired packets that user j has not received/decoded but the other user has. The simple network-coded scheme is described by the following pseudo-code in which the t variable in the FOR-LOOP is the time slot that is currently under consideration and n is an auxiliary variable stored/used by the BS. It is worth noting that we use "packet n" to refer to a packet of index n, while most of the time in the algorithm n is simply an integer variable.

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

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

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2010

UPDATE L_1 and L_2 by the feedback received from the two users at the end of time t.
end for

B. Asymptotic Throughput Optimality of the Proposed Coding Scheme

It is straightforward to show that an upper bound on $\frac{\mathbb{E}\{N_{\text{success}}\}}{2N}$ exists for any N [16]. That is,

$$\frac{\mathbb{E}\{N_{\text{success}}\}}{2N} \leq \min(\lambda p, 1). \tag{5}$$

For the following, we will show that (5) is also achievable by the scheme in Section IV-A.

Equation (5) contains two cases: $0 < p < \frac{1}{\lambda}$ and $\frac{1}{\lambda} \leq p \leq 1$. We first focus on the case in which $p < \frac{1}{\lambda}$. Let $n(t)$ denote the value of the auxiliary variable n in the end of time slot t, which is the index of the next uncoded packet to be sent. Define the "index advancement" at time t as a function of t: $q(t) \triangleq n(t) - \frac{t}{\lambda}$. Note that if $q(t)$ is finite when $N \to \infty$, then we will have $N(t) \approx \frac{t}{\lambda}$. This means that it will take roughly λN slots to finish the transmission of N packets by the proposed policy. Since the proposed scheme only sends packets that are useful to *both users*, the expected reward per user is roughly $\frac{2p\lambda N}{2}$. The proof is thus complete. In the following, we assume $N = \infty$ (continuously streaming with no file-size limit) and use a Lyapunov function to prove that $q(t)$ is indeed bounded away from infinity for $t \in [1, \infty)$ with probability one.

Let $C_j(t_1, t_2) = \sum_{t=t_1+1}^{t_2} C_j(t)$ denote the number of time slots in $(t_1, t_2]$ in which packets successfully arrive user j. We then have the following central lemma, which is also proven in [16].

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

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

We will use Lemma 1 to show that $q(t)$ has a strictly negative drift for sufficiently large $q(t)$, which also implies the boundedness of $q(t)$. This can be done by noting that when $q(t_1) = n(t_1) - \frac{t_1}{\lambda} > B$, we can find a pair $t_2 = t_1 + \lambda B$ and t_1 such that $t_2 < \lambda n(t_1)$. By Lemma 1, we then have

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

For sufficiently large $B = \frac{t_2 - t_1}{\lambda}$, by the law of large numbers, we then have $\mathbb{E}\{\max_{j=1,2} C_j(t_1, t_2)\} \approx p\lambda B$. Therefore

$$\begin{aligned}
&\mathbb{E}\left\{ q(t_1 + \lambda B) - q(t_1) \middle| q(t_1) > B\right\} \\
&= \mathbb{E}\left\{ \left(n(t_2) - \frac{t_2}{\lambda}\right) - \left(n(t_1) - \frac{t_1}{\lambda}\right) \middle| q(t_1) > B \right\} \\
&\approx p\lambda B - \frac{\lambda B}{\lambda} < -\epsilon B,
\end{aligned} \tag{6}$$

where $\epsilon = 1 - p\lambda > 0$. $q(t)$ thus has a negative drift when $q(t) > B$. The negative drift can then be used to show that

$$\frac{\mathbb{E}\{N_{\text{success}}\}}{2N} \geq \frac{2p\mathbb{E}\{t_N\}}{2N} \approx p\lambda$$

As discussed in the beginning of this section, the proof for the case $p < \frac{1}{\lambda}$ is complete.

The detailed proof of Lemma 1 and the expected throughput is in [16]. The case when $p > \frac{1}{\lambda}$ can be proven by similar techniques [16]. In summary the network coding scheme in Section IV-A achieves asymptotically the capacity upper bound in (5).

V. Simulation

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

A. Performance Comparison

Fig. 2 contains four curves: the asymptotic capacity regions for coded and uncoded transmissions and the achievable throughputs for the proposed NC scheme and for the DP non-coding policy with $N = 10000$. The capacity regions (the two curves) are plotted according to the closed form expressions in Sections III and IV-B. The achievable rates for $N = 10000$ are the squares and the triangles, which are plotted by simulation. The predicted asymptotic throughput fits the simulation results perfectly for large $N = 10000$.

In Fig. 3 we compare the NC policy with the uncoded case for small, finite file size N. Even for file size as small as $N = 50$, the NC scheme outperforms the best non-coding DP policy. The expected throughput of the NC scheme deviates slightly more from its asymptotic expression for small N (i.e., $N \leq 100$). That's because, the *index advancement* $q(t)$ is small when t is small, which means that the ongoing packet n is going to expire quickly (with the deadline λn close to t). Due to the randomness of the channel, those initial packets are more likely to expire, which affects the throughput. For larger N (say $N > 100$), the initial loss when t is small is compensated by the success when t is large. The NC scheme thus recovers the capacity loss.

B. Relevance of $q(t)$ with Delay

Fig. 4 shows the time evolution of the index advancement $q(t) = n(t) - \frac{t}{\lambda}$ for $p = 0.33$, which contains the trajectories of the $q(t)$ for 40 random realizations. As predicted in Section IV-B, $q(t)$ remains small (≤ 85) for the entire duration $t \in [1, 5000]$. Among 1000 random realizations, only 70 of the $q(t)$ curves have ever been over 85 for all $t \in [1, 5000]$.

In addition to its role in network coded throughput, the index advancement $q(t)$ is also highly relevant to *transmission delay* in the setting of *sequential packet arrival*. More explicitly, instead of the stored-video application, we consider live video for which not all packets are available in the beginning of the broadcast session. Suppose packet n arrives at the BS at time $\lambda n - \Delta$, where $\Delta > 0$ is the time offset between the arrival time at the BS and the deadline λn at the users. This Δ thus represents the *maximum allowable transmission delay* that includes the queueing, propagation, and decoding delays. Note that in the NC protocol, the packet sent at t_0 is generated

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2010

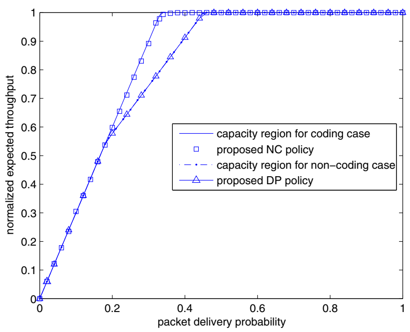

Fig. 2. The capacity curves for the dynamic-programming non-coding policy and for the proposed network coding policy.

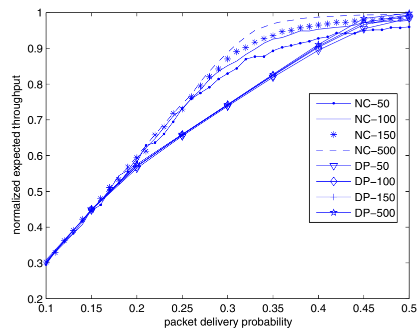

Fig. 3. Performance comparison between NC and DP policies for small finite $N = 50$, 100, 150, and 500.

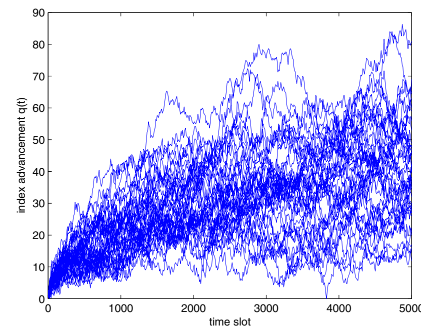

Fig. 4. Time evolution of the index advancement $q(t)$ for $p = 0.33$.

(either codedly or non-codedly) by packets of index $\leq n(t_0)$. If packet $n(t_0)$ has already arrived at the BS by t_0, i.e., if

$$t_0 \geq \lambda n(t_0) - \Delta = \lambda \left(q(t_0) + \frac{t_0}{\lambda} \right) - \Delta \Leftrightarrow \Delta \geq \lambda q(t_0),$$

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

VI. Conclusions

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

In the uncoded case, we have obtained a closed form expression of the optimal throughput. For coded transmission, we have proposed and analyzed a novel network coding (NC) scheme, which achieves the asymptotic capacity without the knowledge about the packet delivery probability. In addition, our simulation shows that the NC scheme achieves strictly higher throughput than that of the best uncoded scheme even for very small file size N. A Lyapunov analysis of the *index advancement* has been developed, which sheds further insight into the NC scheme for both the file-streaming and live-streaming applications. For the case of > 2 users, we have shown by simulation that the throughput of NC scheme is still achieving the asymptotic capacity when N is large. We are currently investigating the rigorous Lyapunov analysis for > 2 users and its corresponding applications for the setting of sequential arrivals and even with *random arrival processes*.

Acknowledgement

This work was supported in part by NSF grants CCF-0845968 and CNS-0721484.

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, February 2003.
[3] R. Koetter and M. Médard, “An algebraic approach to network coding,” *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, November 2003.
[4] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, October 2006.
[5] P. Chou, Y. Wu, and K. Jain, “Practical network coding,” in *Proc. of Allerton Conference*, 2003.
[6] J. Sundararajarn, D. Shah, and M. Médard, “ARQ for network coding,” in *Proc. of ISIT*, July 2008, pp. 1651–1655.
[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] P. Chaporkar and A. Proutiere, “Adaptive network coding and scheduling for maximizing throughput in wireless networks,” in *Proc. of ACM MobiCom*, 2007.
[9] E. Erez, M. Effros, and T. Ho, “Network codes with deadlines,” in *Proc. of Allerton Conference*, 2008.
[10] A. Eryilmaz, A. Ozdaglar, and M. Médard, “On delay performance gains from network coding,” in *Proc. of CISS*, 2006.
[11] J. Barros, R. Costa, D. Munaretto, and J. Widmer, “Effective delay control in online network coding,” in *Proc. of IEEE INFOCOM*, 2009.
[12] J. Sundararajan, D. Shah, and M. Médard, “Online network coding for optimal throughput and delay – the three-receiver case,” in *Arxiv preprint arXiv*, 2008.
[13] W. Yeow, A. Hoang, and C. Tham, “Minimizing delay for multicast-streaming in wireless networks with network coding,” in *Proc. of IEEE INFOCOM*, 2009.
[14] V. Raghunathan, V. Borkar, M. Cao, and P. Kumar, “Index policies for real-time multicast scheduling for wireless broadcast systems,” in *Proc. of IEEE INFOCOM*, 2008.
[15] D. Nguyen, T. Nguyen, and B. Bose, “Wireless broadcast using network coding,” in *NetCod*, 2007.
[16] X. Li, C.-C. Wang, and X. Lin, “Throughput and delay analysis on uncoded and coded wireless broadcast with hard deadline constraints,” *Technical Report, Purdue University, http://web.ics.purdue.edu/∼li179/*, 2009.