

Broadcasting Delay-Constrained Traffic over Unreliable Wireless Links with Network Coding

I-Hong Hou

Abstract—There is increasing demand for using wireless networks for applications that generate packets with strict per-packet delay constraints. In addition to delay constraints, such applications also have various traffic patterns and require guarantees on throughputs of packets that are delivered within their delay constraints. Furthermore, a mechanism for serving delay-constrained traffic needs to specifically consider the unreliable nature of wireless links, which may differ from link to link. Also, as it is usually infeasible to gather feedback information from all clients after each transmission, broadcasting delay-constrained traffic requires addressing the challenge of the lack of feedback information.

We study a model that jointly considers the application requirements on traffic patterns, delay constraints, and throughput requirements, as well as wireless limitations, including the unreliable wireless links and the lack of feedback information. Based on this model, we develop a general framework for designing feasibility-optimal broadcasting policies that applies to systems with various network coding mechanisms. We demonstrate the usage of this framework by designing policies for three different kinds of systems: one that does not use network coding, one that employs XOR coding, and the last that allows the usage of linear coding.

Index Terms—Wireless networks; real-time communications; delay constraints; broadcast; multicast; network coding

I. INTRODUCTION

With the emergence of wireless technologies and mobile devices, it is anticipated that there will be a growing need for using wireless networks for serving delay-constrained flows, such as VoIP and multimedia streaming. Delay-constrained flows have a strict per-packet delay bound on each packet they generate, and packets are not useful if they are delivered after their delay bounds. In this paper, we study systems where a basestation is broadcasting a number of delay-constrained flows to its wireless clients.

One major challenge for broadcast traffic in wireless networks is that wireless links are usually unreliable, and their qualities differ from client to client. One common solution to guarantee reliable deliveries over unreliable links is Automatic Repeat reQuest (ARQ), where clients provide feedback information to the basestation after each transmission using either acknowledgements (ACKs) or negative acknowledgements (NACKs). However, as the

overhead of gathering feedback information increases with the size of the network, using ARQ for wireless broadcasting can incur significant delay and is not scalable. Thus, a solution for wireless broadcasting that does not require feedback information is needed.

Recent work has shown that the capacity of wireless broadcasting over unreliable links can be increased by employing network coding [1]. However, very little research has been conducted on using network coding for delay-constrained traffic. In particular, schemes that require the basestation to obtain feedback information frequently [2]–[4] are not practical except for small-sized networks.

In this paper, we propose an analytical model based on previous studies [5]–[7] that jointly considers the requirements of delay-constrained flows, the demands of wireless clients, and the limitations of wireless networks. This model characterizes a delay-constrained flow by its pattern of packet generation and its per-packet delay bound. The demands of clients are specified by allowing each client to require a minimum of throughput for each flow. Further, the wireless network is modeled as one where wireless links are unreliable. Two features distinguish this model from that in previous studies [5]–[7]: First, previous studies focus on unicast, where the basestation has feedback information on whether a transmission is successful after each transmission, and the basestation can incorporate this information to make scheduling decisions. On the other hand, this work addresses broadcast, where the basestation has no feedback information after each transmission, and needs to schedule packets without relying on such information. Second, this work allows the option of using different network coding mechanisms.

Based on this model, we propose a general framework for developing scheduling policies for the basestation. We prove a sufficient condition for a scheduling policy to be feasibility-optimal among all policies that can be employed by a wireless system using an arbitrary network coding mechanism. Thus, designing scheduling policies for a particular system can be reduced to finding a policy that satisfies the sufficient condition.

We demonstrate the utility of this general framework by evaluating three different kinds of systems, one that does not employ network coding, one that uses XOR coding, and a third that employs linear coding. For systems without network coding, we propose a simple, greedy online scheduling policy. This policy is proved to be feasibility-optimal using the general framework. We also propose on-line policies for systems with XOR coding

I-Hong Hou is with Department of ECE, Texas A&M University, College Station, Texas 77843-3128, USA. email: ihou@tamu.edu

Part of this work has been presented at the 2011 ACM MobiHoc.

and systems with linear coding, and prove that they are feasibility-optimal for their respective systems, under mild restrictions.

In addition, we implement all the proposed policies using the IEEE 802.11 standard in a simulation environment. We compare our policies against a round-robin policy under various scenarios. Simulation results show that our policies outperform the round-robin policy in all scenarios. Further, they also show that those policies that employ network coding achieve better performance than the feasibility-optimal policy for a system without network coding. These simulation results suggest that network coding can also improve the performance when broadcasting delay-constrained flows even when feedback information is not available.

Finally, we aim to understand whether obtaining per-transmission feedback from each client can improve system performance. Intuitively, when basestation has per-transmission feedback information, it can use this information to make better scheduling decisions, which may result in better performance. On the other hand, the amount of time required to gather feedback information can become a significant overhead that hurt the system performance. To understand whether per-transmission feedback is beneficial, we first develop a feasibility-optimal scheduling policy for systems where the basestation has per-transmission feedback. We then compare the performance of this policy against a policy without feedback information through simulations. Simulation results show that, even when the size of the network is small, obtaining per-transmission feedback results in worse performance.

The rest of the paper is organized as follows. Section II summarizes existing related work. Section III formally introduces our analytical model. The general framework for designing feasibility-optimal policies is developed in Section IV. In Section V, we use the general framework to design an on-line scheduling policy for systems without network coding and prove that the policy is feasibility-optimal. Scheduling policies for systems with XOR coding and systems with linear coding are developed in Section VI and Section VII, respectively. Section VIII proposes a feasibility-optimal policy when the basestation has per-transmission feedback from each client. In Section IX, we present the simulation results. Section X concludes this paper.

II. RELATED WORK

There has been increasing research in providing services for delay-constrained flows using wireless networks. Hou, Borkar, and Kumar [5] have analytically studied the problem of scheduling several delay-constrained unicast flows in unreliable wireless environments. This work has been extended to allow variable-bit-rate flows [6], and various models for wireless links [7]. In this paper, we extend the model in [5] to model broadcast flows. Zhang and Du [8] have proposed a cross-layer design for multimedia broadcast. Raghunathan et al [9] have proposed

scheduling policies for broadcasting delay-constrained flows. This work only focuses on minimizing the total number of expired packets and does not consider the different throughput requirements on different flows for each client. Gopala and El Gamal [10] have studied the tradeoff between throughput and delay of broadcasting. They have only studied the scaling laws of average delay, and thus their results are not applicable to scenarios where strict per-packet delay bounds are required. Zhou and Ying [11] have studied the asymptotic capacity of delay-constrained broadcast in mobile ad-hoc networks. Won et al. [12] have proposed multicast scheduling policies to achieve proportional fairness. They have assumed that wireless transmissions are reliable when a proper data rate is selected, and all clients provide information of their channels to the basestation in every time slot. Deb, Jaiswal, and Nagaraj [13] have studied multicasting real-time video streams using WiMax.

Network coding has emerged as a powerful technique to improve the capacity of wireless networks. Chaporkar and Proutiere [14] have proposed an adaptive network coding policy to improve throughputs of multi-hop unicast flows. Ghaderi, Towsley and Kurose [1] have quantified the reliability gain of network coding for broadcasting in unreliable wireless environments. Nguyen et al [15] have compared the throughputs of broadcast flows in systems employing network coding and those without network coding. Lucani, Medard, and Stojanovic [16] have analyzed the computational overhead of using different network coding schemes. Kozat [17] has studied the throughput capacity when erasure codes are employed. These works focus on throughputs and do not consider delays. Yeow, Hoang, and Tham [18] have focused on minimizing delay for broadcast flows by using network coding. Eryilmaz, Ozdaglar, and Medard [19] have studied the gain in delay performance resulting from network coding. Ying, Yang, and Srikant [20] have demonstrated that coding achieves the optimal delay-throughput trade-off in mobile ad-hoc networks. These works only consider the performance of average delays and do not address strict per-packet delay bounds. Li, Wang, and Lin [3] have studied a special case where a basestation is broadcasting delay-constrained flows to two clients. They demonstrate that using opportunistic network coding achieves the maximum asymptotic throughput for this special case. Both Pu et al. [4] and Gangammanavar and Eryilmaz [2] have studied optimal coding strategies for broadcasting delay-constrained flows. Their works require the basestation to obtain feedback information from clients frequently, and thus may not be scalable. Albanese et al. [21] have proposed Priority Encoding Transmission (PET) that offers different reliability for different packets. This approach requires computing the priority of packets in advance.

III. SYSTEM MODEL

We consider a wireless system where there is one basestation broadcasting several flows with delay constraints to

a number of wireless clients. We denote by \mathbb{I} the set of flows, and by \mathbb{N} the set of clients.

We extend the model proposed in [5] and [6], which considers unicast flows with delay constraints, to model both delay-constrained broadcast flows and unreliable wireless links. We assume that time is slotted and numbered as $\tau \in \{0, 1, 2, \dots\}$. The basestation can make exactly one transmission in a time slot and the duration of a time slot is hence set to be the time needed for broadcasting one packet. Time slots are divided into *intervals* where each interval consists of the T consecutive time slots in $[kT, (k+1)T)$, for all $k \geq 0$. According to its specific traffic pattern, each flow may generate at most one packet at the beginning of each interval. We model the traffic patterns of flows as an irreducible Markov chain with finite states and assume that, in the steady state, the probability that the subset $S \subseteq \mathbb{I}$ of flows generates packets is $R(S)$. Each packet generated by any flow has a delay constraint of T time slots; that is, it needs to be delivered to its client within the same interval that it is generated. At the end of each interval, packets generated at the beginning of that interval expire, and are dropped from the system. Thus, the delay undergone by successfully delivered packet is at most T time slots.

We consider heterogeneous and unreliable wireless links. We assume that the reliability of the channel between the basestation and client n is p_n . When the basestation broadcasts a packet, client n receives that packet correctly with probability p_n , and the packet is corrupted due to channel unreliability with probability $1 - p_n$. Since the overhead of gathering feedbacks from clients is large for broadcast, and ACKs are not implemented for broadcast in most mechanisms, we assume that the basestation has no knowledge of whether clients receive the packet correctly after each transmission. The lack of feedback information is one of the most important characteristics that distinguishes our model from those used in [5] and [6] for unicast flows. While it is infeasible for the basestation to gather feedbacks from clients on a per-transmission basis, the basestation can still obtain feedbacks infrequently. Such infrequent feedbacks are used to estimate the channel condition for each client, rather than to acknowledge the reception of a packet. Thus, we assume that the basestation has knowledge of the channel reliabilities p_n for all $n \in \mathbb{N}$. Since wireless links are unreliable, the basestation may need to broadcast the same packet more than once in an interval to increase the probability of delivery, and thus it is possible that a client receives duplicate packets. In such a case, the duplicate packets are dropped by the client.

It is usually impossible to guarantee successful delivery of all packets over unreliable wireless links. However, most applications that generate delay-constrained flows can tolerate a small fraction of lost packets. The performance of client n on flow i is measured by the long-term average number of packets of flow i received by client n per interval, which we call the *timely-throughput* of client n on flow i . Further, we assume that each client n has

a specified timely-throughput requirement $q_{i,n}$ for each flow i . The system is considered *fulfilled* if the expected long-term average number of packets from every flow $i \in \mathbb{I}$ received by every client $n \in \mathbb{N}$, excluding duplicate packets, per interval, is at least $q_{i,n}$. Since the steady-state probability that flow i generates a packet in an interval is $\sum_{S:i \in S} R(S)$, the timely-throughput requirement $q_{i,n}$ is equivalent to requiring that at least a fraction $q_{i,n} / \sum_{S:i \in S} R(S)$ of the packets generated by flow i are delivered to client n .

The goal of this paper is to design *feasibility-optimal* broadcast policies which fulfill all *strictly feasible* systems. Since the set of broadcast policies depends on the coding mechanism used by the system, we also need to define the concept of *schedule space*.

Definition 1: A *schedule space* of a coding mechanism is the collection of all schedules for an interval. It consists of, for each $S \subseteq \mathbb{I}$, the decision of what packet to transmit in each of time slots within the interval, that can be carried out by the coding mechanism, given that only flows in S generate a packet at the beginning of the interval.

Definition 2: A *broadcast policy* is one that, based on past system history and packet generations in the current interval, assigns a schedule, possibly at random, from the schedule space of its employed coding mechanism.

In this paper, we consider three schedule spaces, one that only transmits raw packets without coding, one that employs XOR coding, and one that employs linear coding.

Definition 3: A system is *strictly feasible* for a schedule space if there exists a positive number $\epsilon > 0$, and a broadcast policy under the schedule space of its coding mechanism, that fulfills the same system with timely-throughput requirement $[(1 + \epsilon)q_{i,n}]$.

Definition 4: A broadcast policy is a *feasibility-optimal* policy under the schedule space of some coding mechanism if it fulfills all systems that are strictly feasible under the use of that coding mechanism.

IV. A FRAMEWORK FOR DESIGNING FEASIBILITY-OPTIMAL POLICIES

We now introduce a framework for designing feasibility-optimal policies under any schedule space. Since the basestation does not have feedback information from clients, it cannot know the actual timely-throughput received by each client for each flow. However, with the knowledge of channel reliabilities, the basestation can estimate the timely-throughputs by calculating the probability that a client obtains the packet of a flow in each interval. Let $e_{i,n}(k)$ be the indicator function that client n actually receives the packet from flow i in the interval $[kT, (k+1)T)$ under some policy η . Note that $e_{i,n}(k)$ is a random variable whose value is not known to the server. Let $\hat{q}_{i,n}(k) := E[e_{i,n}(k) | \mathcal{H}_{kT}]$, where \mathcal{H}_{kT} is the history of all packet arrivals of all the flows up to and including time kT , with the conditional expectation taken under the broadcast policy used by the basestation. Since the probability of successful reception of a packet

by a client depends only on the number of times that a packet is broadcast and conditionally independent of everything else, it follows that $\hat{q}_{i,n}(k)$ is the conditional probability estimate made by the basestation, of whether a packet of flow i is successfully delivered to client n in that interval, based on its actions in that interval. We will denote this aforesaid set of actions of the basestation by \mathcal{A}_{kT} . So $\hat{q}_{i,n}(k) := E[e_{i,n}(k)|\mathcal{H}_{kT}] = E[e_{i,n}(k)|\mathcal{A}_{kT}]$. The random variables $(\hat{q}_{i,n}(k) - e_{i,n}(k))$ are bounded and $E[\hat{q}_{i,n}(k) - e_{i,n}(k)|\mathcal{H}_{kT}] = 0$, for all i, n , and k . Define $e_{i,n} := \liminf_{K \rightarrow \infty} \frac{\sum_{k=0}^{K-1} e_{i,n}(k)}{K}$ and $\hat{q}_{i,n} := \liminf_{K \rightarrow \infty} \frac{\sum_{k=0}^{K-1} \hat{q}_{i,n}(k)}{K}$. The former is the actual long-term timely-throughput of client n on flow i , while the latter is the asymptotic estimate made by the basestation. We then have $\hat{q}_{i,n} = e_{i,n}$ almost surely, by the martingale stability theorem [22, Theorem B, page 458], and thus a system is fulfilled if and only if $\hat{q}_{i,n} \geq q_{i,n}$ for all i and n .

We define the *expected delivery debt* for each client n and flow i as $d_{i,n}(k) := \sum_{j=0}^{k-1} (q_{i,n} - \hat{q}_{i,n}(j))$, that is, the difference between the number of packets of flow i that should have been delivered to client n to fulfill its timely-throughput requirement, and the expected number of packets of flow i that are delivered to client n , up to time kT , as estimated by the basestation. Denote by $D(k)$ the vector consisting of all the expected delivery debts $[d_{i,n}(k)]$. We then have

Lemma 1: A system is fulfilled by a policy η if, under η , $\limsup_{k \rightarrow \infty} E[(\frac{d_{i,n}(k)}{k})^+] = 0$, for all $i \in \mathbb{I}$ and $n \in \mathbb{N}$, where we define $x^+ := \max\{x, 0\}$.

We now provide a sufficient condition for a policy to be feasibility-optimal, similar to that used by Hou and Kumar [7] where unicast flows are considered. The proof is based on the following theorem.

Lemma 2 (Telescoping Lemma): Let $L(t)$ be a non-negative Lyapunov function depending only on \mathcal{F}_t , which denotes the set of all events in the system up to and including time t , i.e., $L(t)$ is adapted to \mathcal{F}_t . Suppose there exist positive constants $B > 0$, $\delta > 0$, and a stochastic process $f(t)$ also adapted to \mathcal{F}_t , such that:

$$E[L(t+1) - L(t)|\mathcal{F}_t] \leq B - \delta E[f(t)|\mathcal{F}_t], \quad (1)$$

then $\limsup_{K \rightarrow \infty} \frac{1}{K} \sum_{t=0}^{K-1} E[f(t)] \leq B/\delta$.

Theorem 1: Let S_k be the set of flows that generate packets in the interval $[kT, (k+1)T)$. A basestation policy η^0 that maximizes

$$\sum_{i \in \mathbb{I}, n \in \mathbb{N}} d_{i,n}(k)^+ \hat{q}_{i,n}(k) \quad (2)$$

for all k , among all policies under its schedule space, is feasibility-optimal for its schedule space.

Proof: Consider a strictly feasible system with timely-throughput requirements $[q_{i,n}]$. There exists a positive number ϵ and a stationary randomized scheduling policy η' , which chooses a schedule randomly from the schedule space, based on the packet arrivals at the beginning of this interval and independent of the system history before

this interval, that fulfills the same system with timely-throughput requirements $[(1 + \epsilon)q_{i,n}]$. Since we model packet generations in each interval as an irreducible finite-state Markov chain and η' is a stationary randomized policy, there exists a large enough positive number M such that the expected average timely-throughputs under η' in any M consecutive intervals is larger than $(1 + \frac{\epsilon}{2})q_{i,n}$, i.e.,

$$E[\frac{\sum_{l=k}^{k+M-1} \hat{q}_{i,n}(l)}{M} | S_k, D(k)] > (1 + \frac{\epsilon}{2})q_{i,n}, \quad (3)$$

for all i, n, S_k , and $D(k)$.

Define $L(t) := \frac{1}{2} \sum_{i \in \mathbb{I}, n \in \mathbb{N}} (d_{i,n}(tM)^+)^2$. We then have

$$L(t+1) = \frac{1}{2} \sum_{i \in \mathbb{I}, n \in \mathbb{N}} [(d_{i,n}(tM)^+ + Mq_{i,n} - \sum_{k=tM}^{(t+1)M-1} \hat{q}_{i,n}(k))^+]^2,$$

and

$$\begin{aligned} & E[L(t+1) - L(t)|\mathcal{H}_{tM}] \\ &= E[L(t+1) - L(t)|S_{tM}, D(tM)] \\ &\leq E[\sum_{i \in \mathbb{I}, n \in \mathbb{N}} (Mq_{i,n} - \sum_{k=tM}^{(t+1)M-1} \hat{q}_{i,n}(k))d_{i,n}(tM)^+ \\ &\quad + B_0 | S_{tM}, D(tM)] \quad (4) \\ &= E\{\sum_{k=tM}^{(t+1)M-1} E[\sum_{i,n} d_{i,n}(k)^+ (q_{i,n} - \hat{q}_{i,n}(k)) | S_k, D(k)] \\ &\quad + B_1(\eta) | S_{tM}, D(tM)\}, \quad (5) \end{aligned}$$

where B_0 is a positive constant and $B_1(\eta)$ is bounded by $|B_1(\eta)| < B_2$, for some $B_2 > 0$, regardless of $S_{tM}, D(tM)$, and the employed policy η , because $|d_{i,n}(k) - d_{i,n}(tM)| \leq M$, for all $k \in [tM, (t+1)M)$, and all $i \in \mathbb{I}, n \in \mathbb{N}$.

Let $\hat{q}_{i,n}^0(k)$ and $\hat{q}'_{i,n}(k)$ be the values of $\hat{q}_{i,n}(k)$ under the policies η^0 and η' , respectively. Since η^0 maximizes

$$\sum_{i \in \mathbb{I}, n \in \mathbb{N}} d_{i,n}(k)^+ \hat{q}_{i,n}(k),$$

we have that, under η^0 ,

$$\begin{aligned} & E[L(t+1) - L(t) | S_{tM}, D(tM)] \\ &\leq E\{\sum_{k=tM}^{(t+1)M-1} E[\sum_{i,n} d_{i,n}(k)^+ (q_{i,n} - \hat{q}_{i,n}^0(k)) | S_k, D(k)] \\ &\quad + B_1(\eta^0) | S_{tM}, D(tM)\} \quad (\text{by (5)}) \\ &\leq E\{\sum_{k=tM}^{(t+1)M-1} E[\sum_{i,n} d_{i,n}(k)^+ (q_{i,n} - \hat{q}'_{i,n}(k)) | S_k, D(k)] \\ &\quad + B_1(\eta^0) | S_{tM}, D(tM)\} \\ &\leq E[M \sum_{i \in \mathbb{I}, n \in \mathbb{N}} (q_{i,n} - \frac{\sum_{k=tM}^{(t+1)M-1} \hat{q}'_{i,n}(k)}{M}) d_{i,n}(tM)^+ \\ &\quad + B_1(\eta^0) + B_0 - B_1(\eta') | S_{tM}, D(tM)] \quad (\text{by (5)}) \\ &< -\frac{M\epsilon q^*}{2} \sum_{i \in \mathbb{I}, n \in \mathbb{N}} d_{i,n}(tM)^+ + B, \end{aligned}$$

where $q^* := \min_{i,n:q_{i,n}>0} q_{i,n}$ and $B := 2B_2 + B_0$. The last inequality follows from (3).

By Lemma 2, we have that

$$\limsup_{K \rightarrow \infty} \frac{1}{K} \sum_{t=0}^{K-1} E\left\{ \sum_{i \in \mathbb{I}, n \in \mathbb{N}} d_{i,n}(tM)^+ \right\} \leq \frac{2B}{M \epsilon q^*}.$$

Lemma 4 in [7] shows that $\limsup_{k \rightarrow \infty} E\left[\left(\frac{d_{i,n}(k)}{k}\right)^+\right] = 0$ and the system is also fulfilled by the policy η^0 . Thus, η^0 is feasibility-optimal. ■

Theorem 1 provides an avenue for designing scheduling policies. For any system and any coding mechanism, we can design a policy that aims to maximize (2). Such a policy is feasibility-optimal.

V. SCHEDULING WITHOUT NETWORK CODING

Next, we consider three different kinds of coding mechanisms and show that Theorem 1 suggests tractable scheduling policies. We first consider a system where network coding is not employed. In each time slot, the basestation can only broadcast a raw packet from a flow that has generated one packet in the interval.

Suppose some subset of flows S_k have generated packets at the beginning of the interval $[kT, (k+1)T)$ and that the packet from flow i , $i \in S_k$, is transmitted t_i times within the interval. The probability that client n receives the packet from flow i in this interval is then $\hat{q}_{i,n}(k) = 1 - (1 - p_n)^{t_i}$. Since the basestation can make T broadcasts in an interval, we have $\sum_{i \in S_k} t_i \leq T$. We can then formulate the condition in Theorem 1 as an integer programming problem:

$$\begin{aligned} & \max \sum_{i \in S_k} \sum_n d_{i,n}(k)^+ [1 - (1 - p_n)^{t_i}] \\ & \text{s.t.} \quad \sum_{i \in S_k} t_i \leq T, \\ & \quad t_i \geq 0, \forall i \in S_k. \end{aligned}$$

We show that there exists a polynomial time algorithm that solves the integer programming problem. Suppose that, at some time in an interval, the packet of flow i has been broadcast $t_i - 1$ times. The probability that client n has not received the packet from flow i during the first $t_i - 1$ transmissions, and receives this packet when the basestation broadcasts the packet from flow i for the t_i^{th} time, is $p_n(1 - p_n)^{t_i - 1}$. Thus, we can define the *weighted marginal delivery probability* of the t_i^{th} transmission of flow i as

$$m_i(t_i) := \sum_{n \in \mathbb{N}} d_{i,n}(k)^+ p_n (1 - p_n)^{t_i - 1}.$$

We now propose an online scheduling algorithm, which we call the Greedy Algorithm, as shown in Algorithm 1. In Step 7 of the algorithm, we break ties randomly. We also show that this algorithm is feasibility-optimal.

Theorem 2: Algorithm 1 is feasibility-optimal when network coding is not employed.

Algorithm 1 Greedy Algorithm

```

1: Number flows as  $1, 2, \dots, |\mathbb{I}|$ 
2: for  $i = 1$  to  $|\mathbb{I}|$  do
3:    $t_i \leftarrow 1$ 
4:    $m_i \leftarrow \sum_{n \in \mathbb{N}} d_{i,n}(k)^+ p_n$ 
5: end for
6: for  $\tau = 1$  to  $T$  do
7:    $i \leftarrow \arg \max_{j \in S_k} m_j$ 
8:    $t_i \leftarrow t_i + 1$ 
9:    $m_i \leftarrow \sum_{n \in \mathbb{N}} d_{i,n}(k)^+ p_n (1 - p_n)^{t_i - 1}$ 
10: end for
11: for  $i = 1$  to  $|\mathbb{I}|$  do
12:   for  $\tau = 1$  to  $t_i$  do
13:     broadcast the packet from flow  $i$ 
14:   end for
15:   for  $n \in \mathbb{N}$  do
16:      $d_{i,n}(k+1) \leftarrow d_{i,n}(k) + q_{i,n} - [1 - (1 - p_{b_{i,n}})^{t_i}]$ 
17:   end for
18: end for

```

Proof: Suppose that in some interval $[kT, (k+1)T)$, Algorithm 1 schedules the packet from flow i for transmission t_i^0 times. Suppose there is another algorithm that schedules the packet from flow i for transmission t'_i times, with $\sum_{i \in S_k} t'_i \leq T$. We show that

$$\begin{aligned} & \sum_{i \in S_k, n} d_{i,n}(k)^+ [1 - (1 - p_n)^{t_i^0}] \\ &= \sum_{i \in S_k, n} \sum_{t=1}^{t_i^0} m_i(t) \\ &\geq \sum_{i \in S_k, n} \sum_{t=1}^{t'_i} m_i(t) \\ &= \sum_{i \in S_k, n} d_{i,n}(k)^+ [1 - (1 - p_n)^{t'_i}]. \end{aligned}$$

If $t'_i \leq t_i^0$, for all $i \in S_k$, then the inequality

$$\sum_{i \in S_k, n} \sum_{t=1}^{t_i^0} m_i(t) \geq \sum_{i \in S_k, n} \sum_{t=1}^{t'_i} m_i(t)$$

holds. If there exists some i such that $t'_i > t_i^0$, then there also exists some j such that $t'_j < t_j^0$, since $\sum_{i \in S_k} t'_i \leq T = \sum_{i \in S_k} t_i^0$. By the design of the algorithm and the fact that both $m_i(\cdot)$ and $m_j(\cdot)$ are decreasing functions, we have that $m_j(t'_j) \leq m_i(t_i^0 + 1) \leq m_j(t_j^0) \leq m_j(t'_j + 1)$. Thus, we can decrement t'_j by 1 and increment t'_i by 1 without decreasing the value of $\sum_{i \in S_k, n} \sum_{t=1}^{t'_i} m_i(t)$. We repeat this procedure until $t'_i \leq t_i^0$, for all i , and deduce that $\sum_{i \in S_k, n} \sum_{t=1}^{t_i^0} m_i(t) \geq \sum_{i \in S_k, n} \sum_{t=1}^{t'_i} m_i(t)$. ■

We now analyze the complexity of the Greedy Algorithm. We can implement the Greedy Algorithm using a max-heap, where there is one node for each flow i whose value is m_i . In Steps 4 and 8, it takes $O(|\mathbb{N}|)$ time to compute m_i . It takes $O(|\mathbb{I}| \log |\mathbb{I}|)$ time to construct the

max-heap. In each iteration between Steps 5 and 8, it takes $O(\log |\mathbb{I}|)$ time to find $\arg \max_{j \in S_k} m_j$ and remove the node from the max-heap. It also takes $O(\log |\mathbb{I}|)$ time to insert that node back into the max-heap once its value is updated. Thus, the total complexity of computing the schedule in an interval is $O(|\mathbb{I}| \log |\mathbb{I}| + T|\mathbb{N}| + T \log |\mathbb{I}|)$.

VI. BROADCASTING WITH PAIRWISE XOR CODING

In this section, we address the use of pairwise XOR coding for broadcasting. We assume that the basestation can either broadcast a raw packet from a flow, or it can choose to broadcast an encoded packet (packet from flow $i \oplus$ packet from flow j), the XOR of a packet from flow i with a packet from flow j , which we shall henceforth denote by $i \oplus j$. A client can recover the packet from flow i either upon directly receiving a raw packet from flow i , or upon receiving a raw packet from flow j and an encoded packet $i \oplus j$, for some j . We exhibit a simple example where a system with pairwise XOR coding can achieve strictly better performance than one without network coding.

Example 1: Consider a system with two flows that generate one packet in each interval, and only one client whose channel reliability is $p_1 = 0.5$. Assume that there are six time slots in an interval. Suppose that the basestation transmits each packet three times in an interval. Then we have $\hat{q}_{1,1} = \hat{q}_{2,1} = 0.875$. Thus, a system with timely-throughput requirements $q_{1,1} = q_{2,1} > 0.875$ is not feasible when network coding is not employed. On the other hand, a system that employs pairwise XOR coding can transmit each of the three different types of packets, the raw packet from each flow and the encoded packet $1 \oplus 2$, twice in each interval, which achieves $\hat{q}_{1,1} = \hat{q}_{2,1} = 0.890625$.

We will show in Section VI-A that finding a feasibility-optimal policy with pairwise XOR coding may involve high complexity. Instead, we aim to design a tractable policy with good performance. We impose two restrictions below and find a policy that is feasibility-optimal under the restrictions. We will further show that the performance of this policy is very close to a feasibility-optimal policy without the restrictions in Section VI-A.

Suppose the Greedy Algorithm broadcasts the packet from flow i for a total of t_i^G times in an interval. We sort all flows so that $t_1^G \geq t_2^G \geq \dots$, and enforce the following restrictions on our scheduling policy:

- 1) In addition to raw packets, we only allow encoded packets of the form $(2i - 1) \oplus (2i)$. The intuition behind this restriction is that we only combine two packets which have each been transmitted a similar number of times under the Greedy Algorithm, which implies that they have similar importance.
- 2) The total number of transmissions scheduled for the raw packets from flow $2i - 1$ and flow $2i$, as well as the encoded packet $(2i - 1) \oplus (2i)$, equals $t_{2i-1}^G + t_{2i}^G$. The intuition behind this restriction is that we aim to enhance the performance of flows $2i - 1$ and $2i$ by XOR coding without hurting other flows.

We call the above two restrictions the *orderly combination restriction* and the *transmission conservation restriction for XOR coding*, respectively. Suppose that, under some policy η that follows the above restrictions, the raw packet from flow i is transmitted t_i times, and the encoded packet $(2i - 1) \oplus (2i)$ is transmitted $t_{(2i-1) \oplus (2i)}$ times in the k^{th} interval. The probability that client n receives the packet from flow $(2i - 1)$ is $\hat{q}_{2i-1,n}(k) = 1 - (1 - p_n)^{t_{2i-1}} [(1 - p_n)^{t_{2i}} + (1 - p_n)^{t_{(2i-1) \oplus (2i)}} - (1 - p_n)^{t_{2i} + t_{(2i-1) \oplus (2i)}}]$. Similarly, we also have $\hat{q}_{2i,n}(k) = 1 - (1 - p_n)^{t_{2i}} [(1 - p_n)^{t_{2i-1}} + (1 - p_n)^{t_{(2i-1) \oplus (2i)}} - (1 - p_n)^{t_{2i-1} + t_{(2i-1) \oplus (2i)}}]$. By Theorem 1, designing a feasibility-optimal policy among all policies that follow the above restrictions can be simplified to one of solving the following integer programming problem for all k :

$$\begin{aligned} & \max \sum_{i=1}^{|S_k|/2} \sum_n d_{2i-1,n}(k)^+ \hat{q}_{2i-1,n}(k) + d_{2i,n}(k)^+ \hat{q}_{2i,n}(k) \\ & \text{s.t. } t_{2i-1} + t_{2i} + t_{(2i-1) \oplus (2i)} = t_{2i-1}^G + t_{2i}^G, \forall 1 \leq i \leq |S_k|/2, \\ & \quad t_i \geq 0, \quad \forall 1 \leq i \leq |S_k|, \\ & \quad t_{(2i-1) \oplus (2i)} \geq 0, \quad \forall 1 \leq i \leq |S_k|/2. \end{aligned}$$

In the formulation, we assume that $|S_k|$, the number of flows that generate a packet in the k^{th} interval, is even. If $|S_k|$ is odd, we can add an imaginary flow i^* into the system to make $|S_k|$ even. We set $q_{i^*,n} = 0$, for all n , and thus $t_{i^*}^G = 0$ since it will never be scheduled by the Greedy Algorithm. The condition $t_{2i-1} + t_{2i} + t_{(2i-1) \oplus (2i)} = t_{2i-1}^G + t_{2i}^G$ allows us to decompose this integer programming problem into $|S_k|/2$ subproblems so that the i^{th} subproblem only involves flows $2i - 1$ and $2i$:

$$\begin{aligned} & \max \sum_n d_{2i-1,n}(k)^+ \hat{q}_{2i-1,n}(k) + d_{2i,n}(k)^+ \hat{q}_{2i,n}(k) \\ & \text{s.t. } t_{2i-1} + t_{2i} + t_{(2i-1) \oplus (2i)} = t_{2i-1}^G + t_{2i}^G, \\ & \quad t_{2i-1}, t_{2i}, t_{(2i-1) \oplus (2i)} \geq 0. \end{aligned}$$

If we further assume that $t_{(2i-1) \oplus (2i)}$ is fixed, this subproblem is equivalent to

$$\begin{aligned} & \max \sum_n d_{2i-1,n}(k)^+ (1 - p_n)^{t_{(2i-1) \oplus (2i)}} [1 - (1 - p_n)^{t_{2i-1}}] \\ & \quad + \sum_n d_{2i,n}(k)^+ (1 - p_n)^{t_{(2i-1) \oplus (2i)}} [1 - (1 - p_n)^{t_{2i}}] + C \\ & \text{s.t. } t_{2i-1} + t_{2i} = t_{2i-1}^G + t_{2i}^G - t_{(2i-1) \oplus (2i)}, \\ & \quad t_{2i-1}, t_{2i} \geq 0, \end{aligned}$$

where C is a constant. The optimal (t_{2i-1}, t_{2i}) for this problem can be found by Algorithm 2. The complexity of Algorithm 2 is $O(|\mathbb{N}|(t_{2i-1}^G + t_{2i}^G))$. We have the following lemma, whose proof is essentially the same as that of Theorem 2.

Lemma 3: Given $t_{(2i-1) \oplus (2i)}$, the pair (t_{2i-1}, t_{2i}) found by Algorithm 2 maximizes

$$\sum_n d_{2i-1,n}(k)^+ \hat{q}_{2i-1,n}(k) + d_{2i,n}(k)^+ \hat{q}_{2i,n}(k).$$

Algorithm 2 GreedyXOR($i, t_{(2i-1)\oplus(2i)}, t_{2i-1}^G, t_{2i}^G$)

```

1:  $t_{2i-1} \leftarrow 1$ 
2:  $t_{2i} \leftarrow 1$ 
3:  $m_{2i-1} \leftarrow \sum_{n \in \mathbb{N}} d_{2i-1,n}(k)^+ (1-p_n)^{t_{(2i-1)\oplus(2i)}} p_n$ 
4:  $m_{2i} \leftarrow \sum_{n \in \mathbb{N}} d_{2i,n}(k)^+ (1-p_n)^{t_{(2i-1)\oplus(2i)}} p_n$ 
5: for  $\tau = 1$  to  $t_{2i-1}^G + t_{2i}^G - t_{(2i-1)\oplus(2i)}$  do
6:    $j \leftarrow \arg \max\{m_{2i-1}, m_{2i}\}$ 
7:    $t_j \leftarrow t_j + 1$ 
8:    $m_j \leftarrow \sum_{n \in \mathbb{N}} d_{j,n}(k)^+ (1-p_n)^{t_{(2i-1)\oplus(2i)}} p_n (1-p_n)^{t_j-1}$ 
9: end for
10: return  $(t_{2i-1}, t_{2i})$ 

```

Using Algorithm 2 as a building block, we propose the *Orderly XOR* algorithm, shown in Algorithm 3, to find the optimal schedule when pairwise XOR coding is employed under the two aforementioned restrictions. The complexity of the Orderly XOR algorithm is $O(|\mathbb{I}| \log |\mathbb{I}| + T|\mathbb{N}| + T \log |\mathbb{I}| + \sum_{i=1}^{\lfloor S_k/2 \rfloor} |\mathbb{N}| (t_{2i-1}^G + t_{2i}^G)^2) = O(|\mathbb{I}| \log |\mathbb{I}| + T \log |\mathbb{I}| + T^2 |\mathbb{N}|)$. The following theorem is the direct result of Lemma 3 and Theorem 1.

Theorem 3: The Orderly XOR algorithm is feasibility-optimal among all policies that follow the orderly combination restriction and the transmission conservation restriction for XOR coding. In particular, the Orderly XOR algorithm fulfills every system that can be fulfilled by the Greedy Algorithm.

Algorithm 3 Orderly XOR

```

1: Obtain  $t_1^G, t_2^G, \dots$  from the Greedy Algorithm
2: Sort flows so that  $t_1^G \geq t_2^G \geq \dots$ 
3: for  $i = 1$  to  $\lfloor S_k/2 \rfloor$  do
4:    $Opt \leftarrow -\infty$ 
5:   for  $t = 0$  to  $t_{2i-1}^G + t_{2i}^G$  do
6:      $(t_{2i-1}, t_{2i}) \leftarrow \text{GreedyXOR}(i, t_{(2i-1)\oplus(2i)}, t_{2i-1}^G, t_{2i}^G)$ 
7:     if  $\sum_n d_{2i-1,n}(k)^+ \hat{q}_{2i-1,n}(k) + d_{2i,n}(k)^+ \hat{q}_{2i,n}(k) > Opt$  then
8:        $Opt \leftarrow \sum_n d_{2i-1,n}(k)^+ \hat{q}_{2i-1,n}(k) + d_{2i,n}(k)^+ \hat{q}_{2i,n}(k)$ 
9:        $t_{2i-1}^X \leftarrow t_{2i-1}$ 
10:       $t_{2i}^X \leftarrow t_{2i}$ 
11:       $t_{(2i-1)\oplus(2i)}^X \leftarrow t$ 
12:     end if
13:   end for
14: end for
15: for  $i = 1$  to  $\lfloor S_k/2 \rfloor$  do
16:   for  $\tau = 1$  to  $t_{2i-1}^X$  do
17:     Broadcast the packet from flow  $2i - 1$ 
18:   end for
19:   for  $\tau = 1$  to  $t_{2i}^X$  do
20:     Broadcast the packet from flow  $2i$ 
21:   end for
22:   for  $\tau = 1$  to  $t_{(2i-1)\oplus(2i)}^X$  do
23:     Broadcast the packet  $(2i - 1) \oplus (2i)$ 
24:   end for
25: end for

```

A. Complexity and Performance Gain of Feasibility-Optimal XOR Coding without Restrictions

We impose two restrictions, namely, the orderly combination restriction and the transmission conservation restriction for XOR coding, in deriving Algorithm 3. We now discuss the complexity that can incur without these two restrictions.

We consider an algorithm that exhaustively searches all possible schedules with pairwise XOR coding, and finds one that maximizes $\sum_{i \in S_k} \sum_n d_{i,n}(k)^+ \hat{q}_{i,n}(k)$ in each interval. The complexity of such an algorithm comes from two parts: First, this algorithm needs to find all possible pairings between packets. The number of possible pairings can be written as $\prod_{i=1}^{\lfloor S_k/2 \rfloor} |\mathbb{S}_k|/2(2i-1)$. Second, for each pairing among all packets, the algorithm needs to find the number of time slots allocated to each pair, including both raw packets and coded packets. The number of ways to allocate T time slots among pairs equals the number of ways to divide T identical items into $|\mathbb{S}_k|/2$ different groups, which is $\binom{T+|\mathbb{S}_k|/2-1}{T}$. Therefore, the complexity of such an algorithm can be preventive even for a limited number of flows and clients.

We further compare the performance of the feasibility-optimal policy without the two restrictions against Algorithm 3 through simulations. Due to the high complexity without the two restrictions, we only consider a simple case as follows: We consider a system with only one client and eight flows. We set the channel reliability of the client to be 0.5 and assume that each flow generates one packet in each interval. In each run, we assign the expected debt to each flow uniformly at random, and we obtain the value of $\sum_i \sum_n d_{i,n}(k)^+ \hat{q}_{i,n}(k)$ under the two policies, respectively. We vary the number of time slots in an interval T from 21 to 30, and for each T we run the simulation 100 times. For each T , we compute the ratio between the average of $\sum_i \sum_n d_{i,n}(k)^+ \hat{q}_{i,n}(k)$ under the two policies.

Simulation results are shown in Fig. 1. It can be shown that the ratio is very close to 1. In fact, for all cases, the ratio is at least 0.997. This suggests that the performance of Algorithm 3 is very close to a feasibility-optimal policy without the two restrictions, and it achieves good performance with low complexity.

VII. BROADCASTING WITH LINEAR CODING

In this section, we address the use of linear coding to improve the performance of broadcasting delay-constrained flows. We assume that, in addition to raw packets, the basestation can also broadcast packets that contain linear combinations of packets from any subset of flows $L \subseteq S_k$. A client can decode all packets from the subset L of flows if it receives at least $|L|$ packets that contain linear combinations of packets from these flows. If a client receives less than $|L|$ packets containing such linear combinations, none of the packets from these flows can be decoded. We first exhibit a simple example

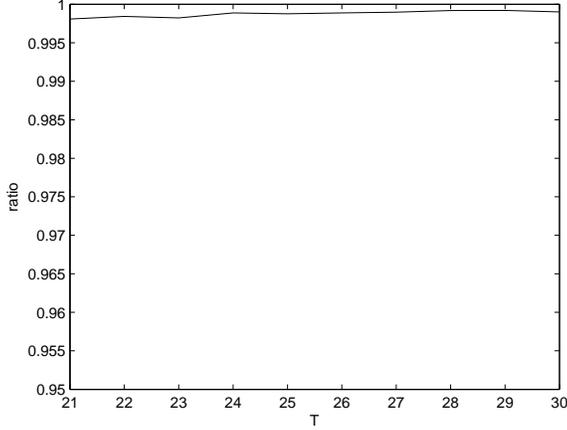


Fig. 1: Performance comparison of pairwise XOR coding with and without the two restrictions.

where a system that uses linear coding provides better performance than one that does not use network coding.

Example 2: Consider a system with one client, whose channel reliability is $p_1 = 0.5$, three flows that generate one packet in each interval, and nine time slots in an interval. A similar argument as that in Example 1 shows that $q_{1,1} = q_{2,1} = q_{3,1} > 0.875$ is not feasible when network coding is not employed. On the other hand, if the basestation employs linear coding and broadcasts a linear combination of the three flows in each time slot, the client can decode all packets from the three flows if it receives at least three packets out of the nine transmissions in an interval, which has probability 0.91015625.

As in Section VI, we address the problem of finding a tractable scheduling policy under some mild restrictions, and then compare its performance against a feasibility-optimal policy without these restrictions. Suppose that the Greedy Algorithm schedules t_i^G transmissions for the packet from flow i in some interval. We sort all flows so that $t_1^G \geq t_2^G \geq \dots$, and enforce the following restrictions:

- 1) Flows are grouped into subsets as $L_1 = \{1, 2, \dots, l_1\}, L_2 = \{l_1 + 1, \dots, l_2\}, \dots$. In each time slot, the basestation broadcasts a linear combination of packets from flows in one of the subsets L_1, L_2, \dots . The intuition behind this restriction is that we only combine packets that have been scheduled similar numbers of times.
- 2) The basestation broadcasts linear combinations of packets from the subset $L_h = \{l_{h-1} + 1, l_{h-1} + 2, \dots, l_h\}$ a total number of $\sum_{i=l_{h-1}+1}^{l_h} t_i^G$ times, where we set $l_0 = 0$. The intuition behind this restriction is that we aim to enhance the performance of flows within L_h without hurting other flows.

The two restrictions above are called the *adjacent combination restriction* and the *transmission conservation restriction for linear coding*, respectively.

We define $r_{n,u,v}$ as the probability that client n receives at least u packets successfully out of v transmissions. We

can compute $r_{n,u,v}$ for all $n \in \mathbb{N}$, $1 \leq u \leq T$, and $1 \leq v \leq T$ in $O(|\mathbb{N}|T^2)$ time using the following iteration:

$$r_{n,u,v} = \begin{cases} 1, & \text{if } u = 0, \\ 0, & \text{if } u > 1, v = 0, \\ p_n r_{n,u-1,v-1} + (1-p_n) r_{n,u,v-1}, & \text{else.} \end{cases}$$

If flows are grouped as L_1, L_2, \dots in the k^{th} interval, the probability that client n is able to obtain the packet from flow $i \in L_h$ is then $\hat{q}_{i,n}(k) = r_{n,|L_h|,(\sum_{i=l_{h-1}+1}^{l_h} t_i^G)}$. We need to find the optimal way to group flows such that

$$\sum_{i \in S_k} \sum_n d_{i,n}(k)^+ \hat{q}_{i,n}(k)$$

is maximized. We solve this problem by dynamic programming. Let $H_{i,j}$ be the optimal way to group flows $i, i+1, \dots, j$, if flows within $[i, j]$ are not allowed to be grouped with flows outside $[i, j]$. We represent $H_{i,j}$ by the collection of groups formed by flows within $[i, j]$. We then have that $H_{i,j}$ either contains one single group consisting of all flows within $[i, j]$, or is of the form $H_{i,h} \cup H_{h+1,j}$, for some $i \leq h < j$. The optimal way to group all flows, which is $H_{1,|S_k|}$, can be found by dynamic programming as in Algorithm 4. The complexity of Algorithm 4 is $O(|\mathbb{I}| \log |\mathbb{I}| + T|\mathbb{N}| + T \log |\mathbb{I}| + |\mathbb{N}||\mathbb{I}|^3)$. We have the following theorem:

Theorem 4: The Optimal Grouping policy, as described in Algorithm 4, is feasibility-optimal among all policies that follow the adjacent combination restriction and the transmission conservation restriction for linear coding. In particular, The Optimal Grouping policy fulfills all systems that can be fulfilled by the Greedy Algorithm.

Proof: The first part of the theorem follows from the discussion in the previous paragraph and Theorem 1. The second part of the theorem follows because a policy that sets $H_{1,|S_k|} = \{\{1\}, \{2\}, \dots\}$ in each interval k has the same schedule as that resulting from the Greedy Algorithm. ■

A. Complexity and Performance Gain of Feasibility-Optimal Linear Coding without Restrictions

We now study the complexity and performance of a feasibility-optimal policy without the two restrictions imposed for Algorithm 4. We consider an algorithm that exhaustively searches all possible schedules with linear coding, and finds one that maximizes $\sum_{i \in S_k} \sum_n d_{i,n}(k)^+ \hat{q}_{i,n}(k)$ in each interval. The complexity of such an algorithm comes from two parts: First, this algorithm needs to find all possible ways to divide packets into arbitrary number of groups. While it does not have closed-form expression, the number of ways to group packets is larger than exponential to the number of packets. Second, for each grouping among all packets, the algorithm needs to find the number of time slots allocated to each group. Therefore, the complexity of such an algorithm can be preventive even for a limited number of flows and clients.

Algorithm 4 Optimal Grouping

```

1: Obtain  $t_1^G, t_2^G, \dots$  from the Greedy Algorithm
2: Sort flows so that  $t_1^G \geq t_2^G \geq \dots$ 
3: for  $i = 1$  to  $|S_k|$  do
4:    $O_{i,i} \leftarrow \sum_n d_{i,n}(k)^+ r_{n,1,t_i^G}$ 
5:    $H_{i,i} \leftarrow \{\{i\}\}$ 
6: end for
7: for  $s = 2$  to  $|S_k|$  do
8:   for  $i = 1$  to  $|S_k| - s + 1$  do
9:      $total \leftarrow \sum_{h=i}^{i+s-1} t_h^G$ 
10:     $O_{i,i+s-1} \leftarrow \sum_{h=i}^{i+s-1} \sum_n d_{h,n}(k)^+ r_{n,s,total}$ 
11:     $H_{i,i+s-1} \leftarrow \{\{i, i+1, \dots, i+s-1\}\}$ 
12:    for  $j = i$  to  $i+s-2$  do
13:      if  $O_{i,j} + O_{j+1,i+s-1} > O_{i,i+s-1}$  then
14:         $O_{i,i+s-1} = O_{i,j} + O_{j+1,i+s-1}$ 
15:         $H_{i,i+s-1} = H_{i,j} \cup H_{j+1,i+s-1}$ 
16:      end if
17:    end for
18:  end for
19: end for
20: Group flows as in  $H_{1,|S_k|}$  and broadcast them accordingly

```

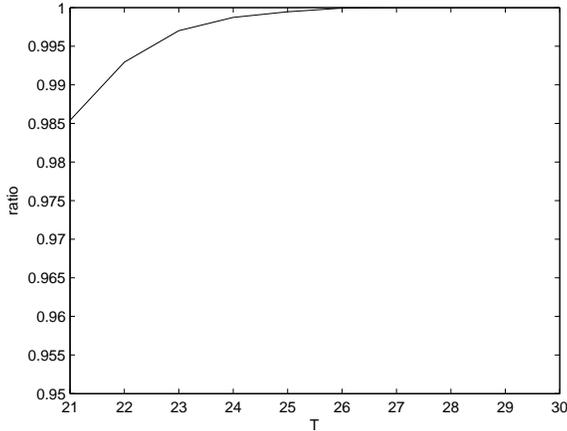


Fig. 2: Performance comparison of linear coding with and without the two restrictions.

We further compare the performance of the feasibility-optimal policy without the two restrictions against Algorithm 4 through simulations. Due to the high complexity without the two restrictions, we only consider a simple case, whose setup is identical to that in Section VI-A. For each T , we compute the ratio between the average of $\sum_i \sum_n d_{i,n}(k)^+ \hat{q}_{i,n}(k)$ under the feasibility-optimal policy without the two restrictions and Algorithm 4.

Simulation results are shown in Fig. 2. It can be shown that the ratio is very close to 1, with the ratio being larger than 0.985 for all cases. This suggests that the performance of Algorithm 4 is very close to a feasibility-optimal policy without the two restrictions, and it achieves good performance with low complexity.

VIII. SCHEDULING WITH FEEDBACK

We have assumed that the basestation does not have instant knowledge on whether a transmission is successfully received by a client. One may wonder whether enforcing per-transmission feedback information can help the basestation make scheduling decisions and result in better performance. To answer this question, we first develop a feasibility-optimal scheduling policy for systems with per-transmission feedback information in this section. We then compare the performance of systems with and without per-transmission feedback information in Section IX.

The model for systems with per-transmission feedback information is very similar to the one proposed in Section III. The only difference is that we now assume that, after each transmission, the basestation immediately receives information on which clients have successfully received the transmission. The basestation can incorporate this information in deciding the packet to be transmitted in the next time slot.

Let $e_{i,n}(k)$ be the indicator function that client n receives the packet from flow i in the interval $[kT, (k+1)T)$ under some policy η . We can define the *actual delivery debt* for each client n and flow i as $d_{i,n}^*(k) := kq_{i,n} - \sum_{j=0}^{k-1} e_{i,n}^*(j)$. We note that, as the basestation has per-transmission feedback information, the basestation knows the values of $e_{i,n}(k)$, and hence can compute the actual delivery debt for all clients and all flows in all intervals.

We can develop a sufficient condition for a scheduling policy to be feasibility-optimal for systems with per-transmission feedback information. The condition is very similar to the one in Theorem 1:

Theorem 5: Let S_k be the set of flows that generate packets in the interval $[kT, (k+1)T)$. A basestation policy η^0 with per-transmission feedback information that maximizes

$$\sum_{i \in I, n \in \mathbb{N}} E\{d_{i,n}^*(k)^+ e_{i,n}(k)\}, \quad (6)$$

for all k , is feasibility-optimal.

Proof: The proof of this theorem is virtually the same as that of Theorem 1, and is hence omitted. ■

We propose a policy for systems with per-transmission feedback information in Algorithm 5. Essentially, the policy schedules the flow i that maximizes $\sum_{n \in \mathbb{N}} p_n d_{i,n}^*(k)^+ 1(n \text{ has not received the packet from } i)$ in every time slot. The policy can be shown to be feasibility-optimal.

Theorem 6: Algorithm 5 is feasibility-optimal for systems with per-transmission feedback information.

Proof: Let $f_{i,n}(\tau, k)$ be the indicator function that client n receives the packet from flow i in the first τ time slots of the interval $[kT, (k+1)T)$. We then have $e_{i,n}(k) = f_{i,n}(T, k)$. Let $D^*(k)$ be the vector consisting of all actual deliver debts $[d_{i,n}^*(k)]$ and $F(\tau, k)$ be the vector consisting of all $[f_{i,n}(\tau, k)]$. Given $D^*(k)$ and $F(\tau, k)$ for some τ and k , a policy η schedules transmissions for the next $T - \tau$ time slots in the interval. Let $V^\eta(D^*(k), F(\tau, k), \tau, S_k)$

Algorithm 5 Scheduling with Feedback

```

1: Number flows as  $1, 2, \dots, \mathbb{I}$ 
2: Number clients as  $1, 2, \dots, \mathbb{N}$ 
3: for  $\tau = 1$  to  $T$  do
4:   for  $i = 1$  to  $\mathbb{I}$  do
5:      $m_i \leftarrow 0$ 
6:     for  $n = 1$  to  $\mathbb{N}$  do
7:       if  $n$  has not received the packet from  $i$  in the
         interval then
8:          $m_i \leftarrow m_i + p_n d_{i,n}^*(k)^+$ 
9:       end if
10:    end for
11:  end for
12:   $i \leftarrow \arg \max_{j \in S_k} m_j$ 
13:  broadcast the packet from flow  $i$ 
14: end for

```

be the expected value of $\sum_{i \in \mathbb{I}, n \in \mathbb{N}} d_{i,n}^*(k)^+ e_{i,n}(k)$ under policy η , given $D^*(k), F(\tau, k), \tau$, and S_k . We have $\sum_{i \in \mathbb{I}, n \in \mathbb{N}} E\{d_{i,n}^*(k)^+ e_{i,n}(k)\} = V^\eta(D^*(k), F(0, k), 0, S_k)$, where we define $f_{i,n}(0, k) \equiv 0$. We claim that Algorithm 5 maximizes $V^\eta(D^*(k), F(\tau, k), \tau, S_k)$, for all $D^*(k), F(\tau, k), \tau$, and S_k . Hence, it maximizes $\sum_{i \in \mathbb{I}, n \in \mathbb{N}} E\{d_{i,n}^*(k)^+ e_{i,n}(k)\}$, and is feasibility-optimal by Theorem 5.

We prove this claim by induction on τ . When $\tau = T - 1$, there is only one time slot left in the interval. Maximizing $V^\eta(D^*(k), F(\tau, k), \tau, S_k)$ is equivalent to maximizing $E\{\sum_{n \in \mathbb{N}, i \in \mathbb{I}} d_{i,n}^*(k)[e_{i,n}(k) - f_{i,n}(\tau, k)]\}$. If the packet of flow i' is scheduled in the last time slot, we have $E\{\sum_{n \in \mathbb{N}, i \in \mathbb{I}} d_{i,n}^*(k)[e_{i,n}(k) - f_{i,n}(\tau, k)]\} = \sum_{n \in \mathbb{N}} p_{i',n} d_{i',n}^*(k)[1 - f_{i',n}(\tau, k)]$. Since Algorithm 5 schedules the flow that maximizes $\sum_{n \in \mathbb{N}} p_{i',n} d_{i',n}^*(k)[1 - f_{i',n}(\tau, k)]$, it maximizes $V^\eta(D^*(k), F(\tau, k), \tau, S_k)$.

Assume that the claim holds for all $t + 1 \leq \tau < T$. We show that the claim also holds when $\tau = t$. Let η' be the policy using Algorithm 5, and η^M be a policy that maximizes $V^\eta(D^*(k), F(t, k), t, S_k)$, respectively. Assume that η' transmits the packet from flow i' , and η^M transmits that from flow i^M , in the $(t + 1)^{th}$ time slot. Let $F'(t + 1, k)$ and $F^M(t + 1, k)$ be the random variables of $F(t + 1, k)$ after the transmissions scheduled in the $(t + 1)^{th}$ time slot by η' and η^M , respectively. First, consider the case when $i' = i^M$. In this case, $F'(t + 1, k) = F^M(t + 1, k)$ on every sample path. By the induction hypothesis, η' maximizes $V^\eta(D^*(k), F'(t + 1, k), t + 1, S_k)$ for all $F'(t + 1, k)$, and hence we have

$$\begin{aligned}
& V^{\eta'}(D^*(k), F(t, k), t, S_k) \\
&= E\{V^{\eta'}(D^*(k), F'(t + 1, k), t + 1, S_k)\} \\
&\geq E\{V^{\eta^M}(D^*(k), F'(t + 1, k), t + 1, S_k)\} \\
&= V^{\eta^M}(D^*(k), F(t, k), t, S_k).
\end{aligned}$$

Next, consider the case $i' \neq i^M$. By the induction hypothesis, η' maximizes $V^\eta(D^*(k), F^M(t + 1, k), t + 1, S_k)$ for all $F^M(t + 1, k)$. Hence, we can modify η^M so that it

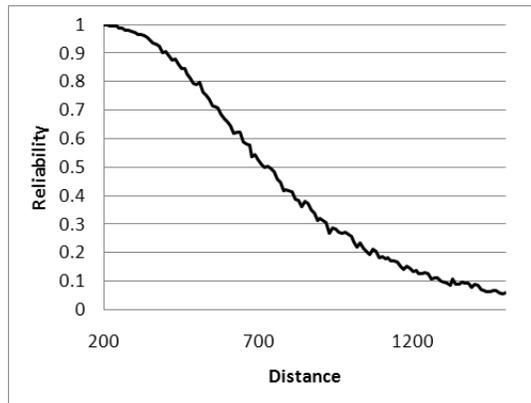


Fig. 3: Relationship between distance and link reliability

schedules i^M in the $(t + 1)^{th}$ time slot, and then follows η' for the remaining time slots in the interval without reducing the value of $V^{\eta^M}(D^*(k), F(t, k), t, S_k)$. As η' schedules i' in the $(t + 1)^{th}$ time slot, i' is the flow that maximizes $\sum_{n \in \mathbb{N}} p_n d_{i,n}^*(k)^+(1 - f_{i,n}(t, k))$. Further, as i' is not scheduled by η^M in the $(t + 1)^{th}$ time slot, $f_{i',n}(t, k) = f_{i',n}(t + 1, k)$ for all n , and i' is the flow that maximizes $\sum_{n \in \mathbb{N}} p_n d_{i,n}^*(k)^+(1 - f_{i,n}(t + 1, k))$. Hence, the modified η^M schedules i' in the $(t + 2)^{th}$ time slot. We can further modify η^M by making it schedules i' in the $(i + 1)^{th}$ time slot, and i^M in the $(i + 2)^{th}$ time slot without reducing the value of $V^{\eta^M}(D^*(k), F(t, k), t, S_k)$. As discussed in the previous paragraph, we now have $V^{\eta'}(D^*(k), F(t, k), t, S_k) \geq V^{\eta^M}(D^*(k), F(t, k), t, S_k)$.

By induction, we have that Algorithm 5 maximizes $V^\eta(D^*(k), F(\tau, k), \tau, S_k)$, for all $D^*(k), F(\tau, k), \tau$, and S_k . ■

IX. SIMULATION RESULTS

We now present our simulation results. We conduct two sets of simulations: one that compares the performance of different policies without per-transmission feedback, and one that compares the performance between policies with and without per-transmission feedback.

A. Systems without Feedback

We have implemented the three scheduling algorithms proposed in this paper, namely, the Greedy Algorithm, the Pairwise XOR algorithm, and the Optimal Grouping algorithm, in ns-2. We compare their performances against a round-robin scheduling policy.

We use the Shadowing module in ns-2 to simulate the unreliable wireless links between the basestation and clients. In the Shadowing module, the link reliability decreases as the distance between two wireless devices increases. The relation between link reliability and distance is shown in Figure 3.

We implement our algorithms based on the IEEE 802.11 standard. Under 802.11, broadcasting a packet with size 160 bytes, which is the size of VoIP packets using the

G.711 codec [23], takes about 2 *ms*. We assume the length of an interval is 40 *ms*, and hence it consists of 20 time slots.

We consider the scenario where a basestation is broadcasting 10 delay-constrained flows to 20 clients that are evenly distributed in a 780×1040 area. We consider two different topologies and timely throughput requirements of clients. The first one is called the *symmetric topology*. In the symmetric topology, the basestation is located at the center of the domain, i.e., at position (390,520). The timely throughput requirements of each client are α for flows 1–5, and β for flows 6–10. That is, we set $q_{i,n} = \alpha$ if $i \leq 5$, and $q_{i,n} = \beta$ if $i > 5$, where α and β are tunable variables to reflect that clients may have different timely throughput requirements for different flows. The other topology that we consider is called the *asymmetric topology*, where the basestation is located at position (520,650). Further, clients in different regions may subscribe to different flows. The timely throughput requirements of flows subscribed to by clients in each region are summarized in Table I. We set $q_{i,n} = 0$ if $q_{i,n}$ does not appear in Table I.

We study two different types of traffic patterns for packet arrivals, namely, *deterministic arrivals* and *probabilistic arrivals*. For deterministic arrivals, we assume that all the 10 flows generate one packet in each interval. This corresponds to flows carrying constant-bit-rate traffic, such as the G.711 codec for VoIP. For probabilistic arrivals, we assume that each flow generates one packet with some probability, independent of the packet generations of other flows, at the beginning of each interval. This scenario corresponds to flows carrying variable-bit-rate traffic. In particular, we assume each of flows 1–5 generates one packet with probability 0.9, and each of flows 6–10 generates one packet with probability 0.6, at the beginning of each interval.

We compare the performances of Algorithms 1, 3, and 4 against two other policies. The first policy is round-robin. The second policy employs linear coding and broadcasts a linear combination of all available packets in each time slot. A client can receive all packets in an interval if and only if the number of transmissions it receives in the interval is at least the number of flows that generates a packet in the interval. This policy is equivalent to MDS(20,10) code. We compare the performances of different scheduling algorithms by comparing the pairs of (α, β) that can be fulfilled by each algorithm. A system is considered to be fulfilled if, after 500 intervals, the average timely throughput of client n on flow i is at least $q_{i,n} - 0.03$.

The simulation results are shown in Figures 4 – 7. As shown in the figures, all the three proposed algorithms outperform the round-robin policy in all scenarios. This is because, without the knowledge of timely throughput requirements, the round-robin policy cannot offer any tradeoff between flows. Simulation results also show that MDS code is always worse than our proposed Optimal Grouping policy. Further, in many cases, the performance

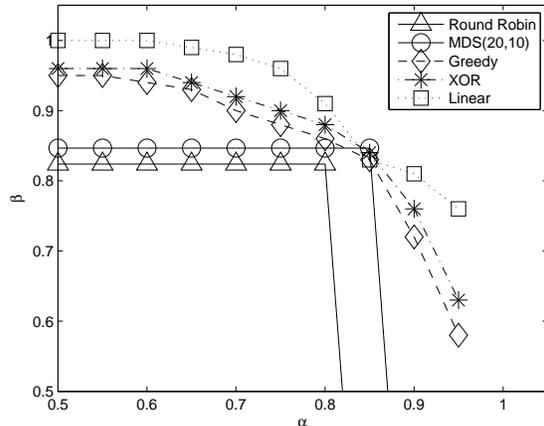


Fig. 4: Simulation results for deterministic arrivals and symmetric topology

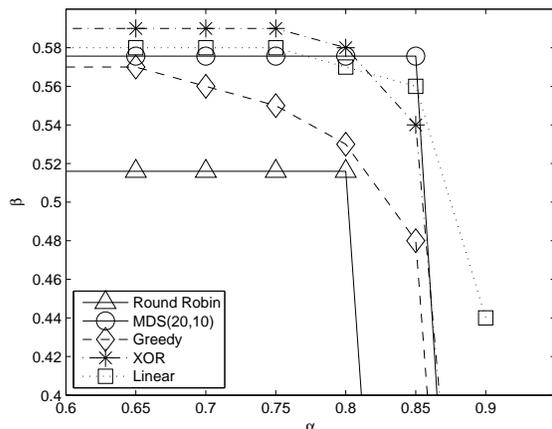


Fig. 5: Simulation results for probabilistic arrivals and symmetric topology

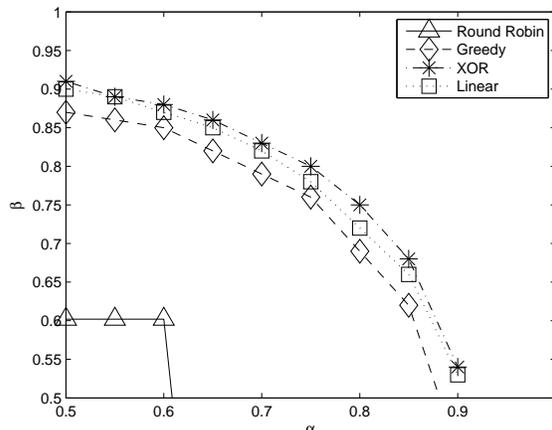


Fig. 6: Simulation results for deterministic arrivals and asymmetric topology

Region	subscribed flows and timely throughput requirements
$[390, 780] \times [520, 1040]$	$q_{1,n} = q_{5,n} = \alpha, q_{6,n} = q_{10,n} = \beta$
$[390, 780] \times [0, 520]$	$q_{2,n} = q_{5,n} = \alpha, q_{7,n} = q_{10,n} = \beta$
$[0, 390] \times [520, 1040]$	$q_{3,n} = q_{5,n} = \alpha, q_{8,n} = q_{10,n} = \beta$
$[0, 390] \times [0, 520]$	$q_{4,n} = q_{5,n} = \alpha, q_{9,n} = q_{10,n} = \beta$

TABLE I: Timely throughput requirements of clients in each region for the asymmetric topology

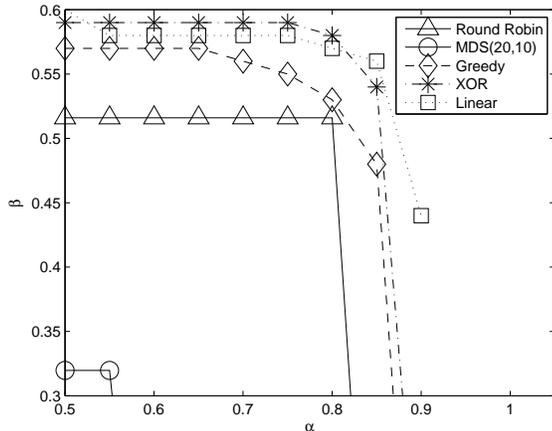


Fig. 7: Simulation results for probabilistic arrivals and asymmetric topology

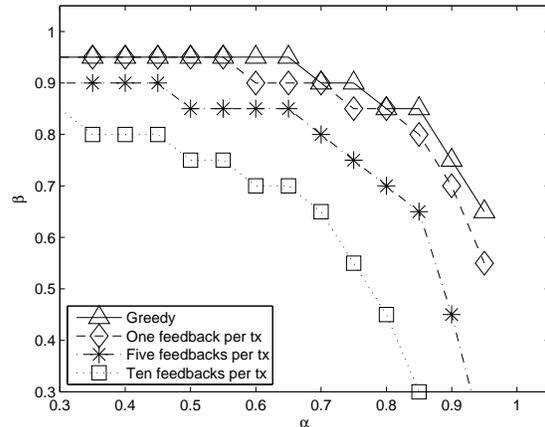


Fig. 8: Performance comparisons between systems with and without per-transmission feedback information.

of MDS code is even worse than round-robin. In particular, in Figure 6, the performance of MDS code is not shown because MDS code even fails to achieve $(0.3, 0.3)$. To understand why MDS code can result in such poor performance, consider a special case where a client has $p_n < 0.5$. Suppose the number of time slots in an interval is twice the number of flows. With round-robin, the packet of each flow is transmitted twice in an interval, and the client has $1 - (1 - p_n)^2$ chance of receiving each packet. On the other hand, with MDS code, the client needs to receive at least half of all transmissions in an interval to obtain all packets, whose probability goes to zero as the number of flow increases.

B. Comparisons Between Systems with and without Feedback

We also compare the performance of systems with and without per-transmission feedback information. We use similar settings as in the previous section. Consider a system with 20 clients and 6 flows. The length of an interval is 40 ms and each flow generates one packet in each interval. We assume that the channel reliability for each client is 60%, and each client requires a timely-throughput of α for the first three flows, and β for the last three flows. We then plot the pairs of all feasible (α, β) for each evaluated system.

For systems without per-transmission feedback information, we assume that there are 20 time slots in an interval, as ns-2 simulations show that it takes 2 ms to broadcast a packet. We apply the Greedy Algorithm for systems without per-transmission feedback information.

We consider that the basestation may employ some compressed sensing technique to obtain whether each client receives a transmission with only a limited number of feedback messages. We assume that the basestation obtains k feedback messages per transmission, where we vary k from 1 to 10, and it can use this information to determine whether each client, including those that do not transmit feedback messages, receives the transmission. Ns-2 simulations show that it takes about 0.13 ms to transmit an ACK. Hence, we assume that, for a system with per-transmission feedback information that has k feedback messages per transmission, it takes $0.13k$ ms to gather feedback information from all clients, and there are $\lfloor \frac{40}{2+0.13k} \rfloor$ time slots in an interval. We apply Algorithm 5 for systems with per-transmission feedback information. We note that, since it is impossible to obtain precise feedback information for all 20 clients with only $k < 20$ feedback messages, this scenario depicts an upper-bound for the performance of all algorithms that employ some intelligent policy in obtaining feedback information.

Figure 8 shows the simulation results. It can be shown that, the Greedy Algorithm outperforms Algorithm 5 even when the basestation only obtains one feedback message per transmission, and the performance difference increases quickly with k . This suggests that the benefits of per-transmission feedback information may actually be much smaller than the cost of obtaining it.

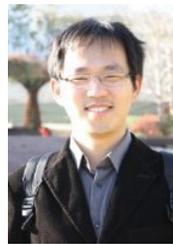
X. CONCLUSION AND FUTURE WORK

We have studied the problem of broadcasting delay-constrained flows in unreliable wireless environments.

We have extended a previous model, which only models unicast flows, to address the additional challenges introduced by broadcasting. This model jointly considers the traffic patterns and delay constraints of flows, the timely throughput requirements of clients, the nature of unreliable wireless links, and the lack of feedback in broadcast. This model also allows the optional usage of various network coding mechanisms. We have proposed a general framework for designing feasibility-optimal scheduling policies. We have demonstrated the usage of this model by developing scheduling policies for three different types of systems, one without network coding, one that employs pairwise XOR coding, and the other using linear coding. Simulation results have shown that the three proposed policies achieve much better performance than a round-robin policy. They also show that policies incorporating network coding have better performance than the feasibility-optimal policy for systems without network coding. This result demonstrates that using network coding can increase the capacity of wireless networks for broadcasting delay-constrained flows.

REFERENCES

- [1] M. Ghaderi, D. Towsley, and J. Kurose, "Reliability gain of network coding in lossy wireless networks," in *Proc. of IEEE INFOCOM*, pp. 2171–2179, 2008.
- [2] H. Gangammanavar and A. Eryilmaz, "Dynamic coding and rate-control for serving deadline-constrained traffic over fading channels," in *Proc. of IEEE ISIT*, pp. 1788–1792, 2010.
- [3] 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 IEEE INFOCOM*, 2010.
- [4] W. Pu, C. Luo, F. Wu, and C. W. Chen, "QoS-driven network coded wireless multicast," *IEEE Transactions on Wireless Communications*, vol. 8, pp. 5662–5670, Nov. 2009.
- [5] I.-H. Hou, V. Borkar, and P. Kumar, "A theory of QoS in wireless," in *Proc. of IEEE INFOCOM*, pp. 486–494, 2009.
- [6] I.-H. Hou and P. Kumar, "Admission control and scheduling for QoS guarantees for variable-bit-rate applications on wireless channels," in *Proc. of ACM MobiHoc*, pp. 175–184, 2009.
- [7] I.-H. Hou and P. Kumar, "Scheduling heterogeneous real-time traffic over fading wireless channels," in *Proc. of IEEE INFOCOM*, 2010.
- [8] X. Zhang and Q. Du, "Cross-layer modeling for QoS-driven multimedia multicast/broadcast over fading channels in mobile wireless networks," *IEEE Communications Magazine*, vol. 45, pp. 62–70, Aug. 2007.
- [9] V. Raghunathan, V. Borkar, M. Cao, and P. Kumar, "Index policies for real-time multicast scheduling for wireless broadcast systems," in *Proceedings of IEEE INFOCOM*, pp. 1570–1578, 2008.
- [10] P. Gopala and H. E. Gamal, "On the throughput-delay tradeoff in cellular multicast," in *Proceedings of the Symposium on Information Theory in WirelessCom*, 2005.
- [11] S. Zhou and L. Ying, "On delay constrained multicast capacity of large-scale mobile ad-hoc networks," in *Proc. of IEEE INFOCOM*, 2010.
- [12] H. Won, H. Cai, D. Y. Eun, K. Guo, A. Netravali, I. Rhee, and K. Sabnani, "Multicast scheduling in cellular data networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 9, pp. 4540–4549, 2009.
- [13] S. Deb, S. Jaiswal, and K. Nagaraj, "Real-time video multicast in wimax networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 1579–1587, 2008.
- [14] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," in *Proceedings of ACM MobiCom*, pp. 135–146, 2007.
- [15] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Transactions on Vehicular Technology*, vol. 58, pp. 914–925, Feb. 2009.
- [16] D. Lucani, M. Medard, and M. Stojanovic, "Systematic network coding for time-division duplexing," in *Proceedings of IEEE ISIT*, pp. 2403–2407, 2010.
- [17] U. Kozat, "On the throughput capacity of opportunistic multicast-ing with erasure codes," in *Proceedings of IEEE INFOCOM*, pp. 520–528, 2008.
- [18] W.-L. Yeow, A. T. Hoang, and C.-K. Tham, "Minimizing delay for multicast-streaming in wireless networks with network coding," in *Proceedings of IEEE INFOCOM*, pp. 190–198, 2009.
- [19] A. Eryilmaz, A. Ozdaglar, and M. Medard, "On delay performance gains from network coding," in *Proc. of CISS*, pp. 864–870, 2006.
- [20] L. Ying, S. Yang, and R. Srikant, "Coding achieves the optimal delay-throughput trade-off in mobile ad-hoc networks: Two-dimensional i.i.d. mobility model with fast mobiles," in *Proc. of WiOpt*, pp. 1–10, 2007.
- [21] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *Information Theory, IEEE Transactions on*, vol. 42, no. 6, pp. 1737–1744, 1996.
- [22] M. Loeve, *Probability Theory*. Litton Educational Publishing, 1963.
- [23] Texas Instruments, "Low power advantage of 802.11a/g vs. 802.11b," Dec. 2003.



I-Hong Hou (S10-M12) received the B.S. in Electrical Engineering from National Taiwan University in 2004, and his M.S. and Ph.D. in Computer Science from University of Illinois, Urbana-Champaign in 2008 and 2011, respectively.

In 2012, he joined the department of Electrical and Computer Engineering at the Texas A&M University, where he is currently an assistant professor. His research interests include wireless networks, wireless sensor networks, real-time systems, distributed systems, and vehicular ad hoc networks.

Dr. Hou received the C.W. Gear Outstanding Graduate Student Award from the University of Illinois at Urbana-Champaign, and the Silver Prize in the Asian Pacific Mathematics Olympiad.