

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

I-Hong Hou
CSL and Department of Computer Science
University of Illinois
Urbana, IL, 61801, USA
ihou2@illinois.edu

P. R. Kumar
CSL and Department of ECE
University of Illinois
Urbana, IL 61801, USA
prkumar@illinois.edu

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.

Categories and Subject Descriptors

C.2.1 [COMPUTER-COMMUNICATION NETWORKS]: Network Architecture and Design —*Wireless communication*

General Terms

Theory

This material is based upon work partially supported by US-ARO under Contract Nos. W911NF-08-1-0238 and W-911-NF-0710287, NSF under Contracts CNS-1035378, CNS-0905397, CNS-1035340, and CCF-0939370, and AFOSR under Contract FA9550-09-0121.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiHoc'11, May 16–20, 2011, Paris, France.

Copyright 2011 ACM 978-1-4503-0722-2/11/05 ...\$10.00.

Keywords

Scheduling, broadcast, deadlines, delays, network coding

1. 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 [4]. 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 [3, 10, 14] are not practical except for small-sized networks.

In this paper, we propose an analytical model 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 and the basestation has no feedback information after each transmission. In addition, this model 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 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.

The rest of the paper is organized as follows. Section 2 summarizes existing related work. Section 3 formally introduces our analytical model. The general framework for designing feasibility-optimal policies is developed in Section 4. In Section 5, we use the general framework to design an online 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 6 and Section 7, respectively. In Section 8, we present the simulation results. Section 9 concludes this paper.

2. RELATED WORK

There has been increasing research in providing services for delay-constrained flows using wireless networks. Hou, Borkar, and Kumar [6] 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 [7], and various models for wireless links [8]. In this paper, we extend the model in [6] to model broadcast flows. Zhang and Du [19] have proposed a cross-layer design for multimedia broadcast. Raghunathan et al [15] 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 [5] have studied the trade-off 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 [20] have studied the asymptotic capacity of delay-constrained broadcast in mobile ad-hoc networks.

Network coding has emerged as a powerful technique to improve the capacity of wireless networks. Chaporkar and Proutiere [1] have proposed an adaptive network coding policy to improve throughputs of multi-hop unicast flows. Ghaderi, Towsley and Kurose [4] have quantified the reliability gain of network coding for broadcasting in unreliable wireless en-

vironments. Nguyen et al [13] have compared the throughputs of broadcast flows in systems employing network coding and those without network coding. Lucani, Medard, and Stojanovic [12] have analyzed the computational overhead of using different network coding schemes. Kozat [9] has studied the throughput capacity when erasure codes are employed. These works focus on throughputs and do not consider delays. Yeow, Hoang, and Tham [17] have focused on minimizing delay for broadcast flows by using network coding. Eryilmaz, Ozdaglar, and Medard [2] have studied the gain in delay performance resulting from network coding. Ying, Yang, and Srikant [18] have demonstrated that coding achieves the optimal delay-throughput tradeoff 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 [10] have studied a special case where a base station 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 [14] and Gangammanavar and Eryilmaz [3] have studied optimal coding strategies for broadcasting delay-constrained flows. Their works require the base station to obtain feedback information from clients frequently, and thus may not be scalable.

3. SYSTEM MODEL

We consider a wireless system where there is one base station 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 [6] and [7], 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 base station 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 base station and client n is p_n . When the base station 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 base station 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 [6] and [7] 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 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 $S \subseteq I$, 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.

4. 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, it can estimate it. Let $q_{i,n}^*(k)$ 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 $q_{i,n}^*(k)$ is a random variable whose value is not known to the server. Let $\hat{q}_{i,n}(k) := E[q_{i,n}^*(k) | \mathcal{H}_{kT}]$, 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 $\hat{q}_{i,n}(k) := E[q_{i,n}^*(k) | \mathcal{H}_{kT}] = E[q_{i,n}^*(k) | \mathcal{A}_{kT}]$. The random variables $(\hat{q}_{i,n}(k) - q_{i,n}^*(k))$ are bounded and $E[\hat{q}_{i,n}(k) - q_{i,n}^*(k) | \mathcal{H}_{kT}] = 0$, for all i, n , and k . Define $q_{i,n}^* := \liminf_{K \rightarrow \infty} \frac{\sum_{k=0}^{K-1} q_{i,n}^*(k)}{K}$ 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 $\hat{q}_{i,n} = q_{i,n}^*$ almost surely, by the martingale stability theorem [11, 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 , $\limsup_{k \rightarrow \infty} (\frac{d_{i,n}(k)}{k})^+ = 0$, 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 [8] 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.,

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

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

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

and

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

$$\begin{aligned} &= E\left\{ \sum_{k=tM}^{(t+1)M-1} E\left[\sum_{i,n} d_{i,n}(k)^+ (q_{i,n} - \hat{q}_{i,n}(k)) \mid S_k, D(k) \right] \right. \\ &\quad \left. + B_1(\eta) \mid S_{tM}, D(tM) \right\}, \end{aligned} \quad (5)$$

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 ,

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

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 Theorem 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 [8] shows that $\limsup_{k \rightarrow \infty} (\frac{d_{i,n}(k)}{k})^+ = 0$ and the system is also fulfilled by the policy η^0 . Thus, η^0 is feasibility-optimal. \square

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.

5. 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:

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

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.

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

```

THEOREM 2. *Algorithm 1 is feasibility-optimal when network coding is not employed.*

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 $m_j(t_j') \leq m_j(t_j^0 + 1) \leq m_j(t_j^0) \leq m_j(t_j' + 1)$. Thus, we can decrement t_j' by 1 and increment t_j^0 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 $\sum_{i \in S_k, n} \sum_{t=1}^{t_i^0} m_i(t) \geq \sum_{i \in S_k, n} \sum_{t=1}^{t_i'} m_i(t)$. \square

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 $O(|\mathbb{I}| \log |\mathbb{I}| + T|\mathbb{N}| + T \log |\mathbb{I}|)$.

6. BROADCASTING WITH XOR CODING

In this section, we address the use of 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 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 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 $\hat{q}_{1,1} = \hat{q}_{2,1} = 0.890625$.*

While it may be computationally complicated to design a feasibility-optimal scheduling policy when XOR coding is employed, we aim to design a tractable policy that achieves better performance than the Greedy Algorithm in Section 5. 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 *pairwise 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 :

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

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 $2i$:

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

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

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

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*

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

Using Algorithm 2 as a building block, we propose the *Pairwise XOR algorithm*, shown in Algorithm 3, to find the optimal

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  $(t_{2i-1}, t_{2i})$ 

```

schedule when XOR coding is employed under the two aforementioned restrictions. The complexity of the Pairwise 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 Pairwise XOR algorithm is feasibility-optimal among all policies that follow the pairwise combination restriction and the transmission conservation restriction for XOR coding. In particular, the Pairwise XOR algorithm fulfills every system that can be fulfilled by the Greedy Algorithm.*

Algorithm 3 Pairwise 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

```

7. 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 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 6, we address the problem of finding a tractable scheduling policy that achieves better performance than the Greedy Algorithm. 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 $O(|\mathbb{N}| \log |\mathbb{N}| + T|\mathbb{N}| + T \log |\mathbb{N}| + |\mathbb{N}||\mathbb{N}|^3)$. 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. \square

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
-

8. SIMULATION RESULTS

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 1.

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

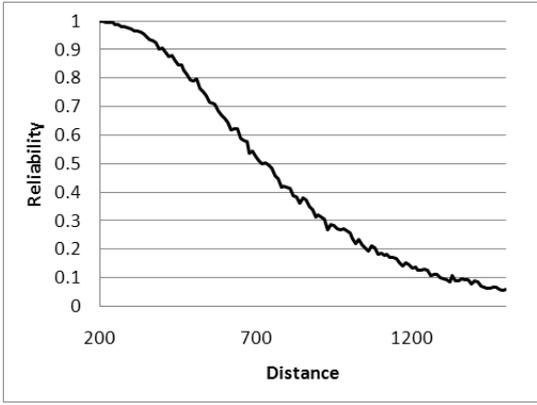


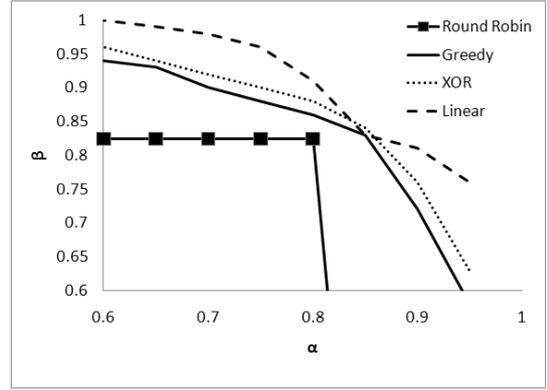
Figure 1: Relationship between distance and link reliability

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 1. We set $q_{i,n} = 0$ if $q_{i,n}$ does not appear in Table 1.

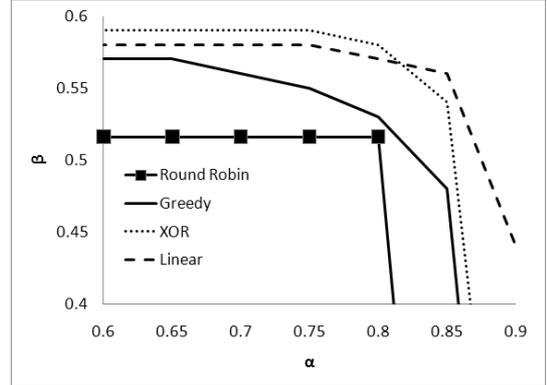
We study two different types of traffic patterns for packet arrivals, namely, *deterministic arrivals* and *probabilistic arrivals*. For deterministic arrivals, we assume that all the 10 flows generate one packet in each interval. This corresponds to flows carrying constant-bit-rate traffic, such as 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, such as MPEG video streaming. 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 evaluate the performances of our algorithms and the round-robin policy for each of the two topologies and each of the two traffic patterns. 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 $q_{i,n} - 0.03$.

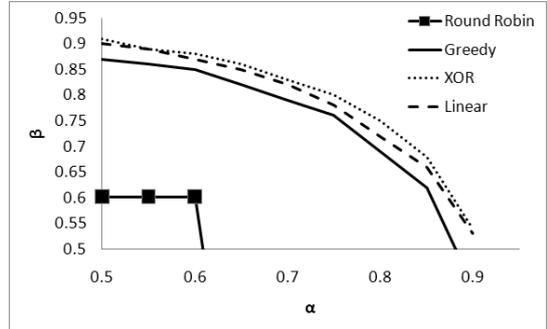
The simulation results are shown in Figure 2. As shown in the figure, all the three proposed algorithms outperform the round-robin policy in all scenarios. This is because, without the knowledge of timely throughput requirements, the round-robin policy cannot offer any tradeoff between flows. Further, the differences of performance between the round-robin policy and the three proposed policies are even more significant



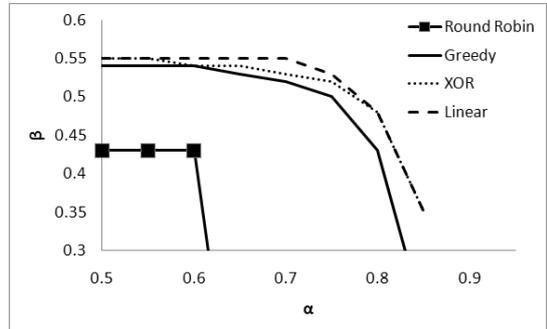
(a) Deterministic arrivals and symmetric topology



(b) Probabilistic arrivals and symmetric topology



(c) Deterministic arrivals and asymmetric topology



(d) Probabilistic arrivals and asymmetric topology

Figure 2: Simulation results

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

in scenarios with asymmetric topology, as shown in Figures 2c and 2d, because the round-robin policy does not incorporate any knowledge of network topology. As the basestation has better links to clients in the region $[390, 780] \times [520, 104]$ than those to clients in the region $[0, 390] \times [0, 520]$, the three proposed policies allocate more transmission opportunities to flows subscribed to by clients in the region $[0, 390] \times [0, 520]$ so as to compensate for their low link qualities. These results show that an efficient policy for broadcasting delay-constrained flows needs to jointly consider both client requirements and network topology. Also, both algorithms using network coding achieve better performance than the Greedy Algorithm, which demonstrates that network coding can be used to increase the capacity of wireless networks for broadcasting delay-constrained flows.

We close this section by comparing the Pairwise XOR algorithm for XOR coding and the Optimal Grouping algorithm for linear coding. In the scenario with deterministic arrivals and symmetric topology, the Optimal Grouping algorithm has much better performance than the Pairwise XOR algorithm. However, the advantage of the Optimal Grouping algorithm becomes less prominent in the other three scenarios, and sometimes it even performs worse than the Pairwise XOR algorithm. The Optimal Grouping algorithm allows combining more than two flows, and thus explores more coding possibilities, which is why it achieves better performance in the first scenario. On the other hand, in systems where the number of generated packets in each interval is less, as in the scenario with probabilistic arrivals, or when the topologies are asymmetric, it becomes less beneficial to combine a large number of packets. In such systems, the Pairwise XOR algorithm may benefit from its simpler coding structure.

9. 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 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.

One important difference between our work and other existing work on broadcasting delay-constrained traffic is that our work does not require frequent feedback from clients. Thus, in large networks, our work should offer much better performance as it avoids the large overhead of gathering feedback. On the other hand, when the size of the network is small, it might be beneficial to obtain feedback from clients and utilize such information. It is an open problem to determine when the price of gathering feedback outweighs the benefit of such information. Also, in the two network coding schemes discussed in this work, we have imposed some simplifying restrictions in order to obtain tractable policies. Whether these restrictions can be relaxed is another challenging problem that requires future research.

10. REFERENCES

- [1] CHAPORKAR, P., AND PROUTIERE, A. Adaptive network coding and scheduling for maximizing throughput in wireless networks. In *Proceedings of ACM MobiCom* (2007), pp. 135–146.
- [2] ERYILMAZ, A., OZDAGLAR, A., AND MEDARD, M. On delay performance gains from network coding. In *Proc. of CISS* (2006), pp. 864–870.
- [3] GANGAMMANAVAR, H., AND ERYILMAZ, A. Dynamic coding and rate-control for serving deadline-constrained traffic over fading channels. In *Proc. of IEEE ISIT* (2010), pp. 1788–1792.
- [4] GHADERI, M., TOWSLEY, D., AND KUROSE, J. Reliability gain of network coding in lossy wireless networks. In *Proc. of IEEE INFOCOM* (2008), pp. 2171–2179.
- [5] GOPALA, P., AND GAMAL, H. E. On the throughput-delay tradeoff in cellular multicast. In *Proceedings of the Symposium on Information Theory in WirelessCom* (2005).
- [6] HOU, I.-H., BORKAR, V., AND KUMAR, P. A theory of QoS in wireless. In *Proc. of IEEE INFOCOM* (2009), pp. 486–494.
- [7] HOU, I.-H., AND KUMAR, P. Admission control and scheduling for QoS guarantees for variable-bit-rate applications on wireless channels. In *Proc. of ACM MobiHoc* (2009), pp. 175–184.
- [8] HOU, I.-H., AND KUMAR, P. Scheduling heterogeneous real-time traffic over fading wireless channels. In *Proc. of IEEE INFOCOM* (2010).
- [9] KOZAT, U. On the throughput capacity of opportunistic multicasting with erasure codes. In *Proceedings of IEEE INFOCOM* (2008), pp. 520–528.
- [10] LI, X., WANG, C.-C., AND LIN, X. Throughput and delay analysis on uncoded and coded wireless

- broadcast with hard deadline constraints. In *Proc. of IEEE INFOCOM* (2010).
- [11] LOEVE, M. *Probability Theory*. Litton Educational Publishing, 1963.
- [12] LUCANI, D., MEDARD, M., AND STOJANOVIC, M. Systematic network coding for time-division duplexing. In *Proceedings of IEEE ISIT* (2010), pp. 2403–2407.
- [13] NGUYEN, D., TRAN, T., NGUYEN, T., AND BOSE, B. Wireless broadcast using network coding. *IEEE Transactions on Vehicular Technology* 58, 2 (2009), 914–925.
- [14] PU, W., LUO, C., WU, F., AND CHEN, C. W. QoS-driven network coded wireless multicast. *IEEE Transactions on Wireless Communications* 8, 11 (2009), 5662–5670.
- [15] RAGHUNATHAN, V., BORKAR, V., CAO, M., AND KUMAR, P. Index policies for real-time multicast scheduling for wireless broadcast systems. In *Proceedings of IEEE INFOCOM* (2008), pp. 1570–1578.
- [16] TEXAS INSTRUMENTS. Low power advantage of 802.11a/g vs. 802.11b.
- [17] YEOW, W.-L., HOANG, A. T., AND THAM, C.-K. Minimizing delay for multicast-streaming in wireless networks with network coding. In *Proceedings of IEEE INFOCOM* (2009), pp. 190–198.
- [18] YING, L., YANG, S., AND SRIKANT, R. 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* (2007), pp. 1–10.
- [19] ZHANG, X., AND DU, Q. Cross-layer modeling for QoS-driven multimedia multicast/broadcast over fading channels in mobile wireless networks. *IEEE Communications Magazine* 45, 8 (2007), 62–70.
- [20] ZHOU, S., AND YING, L. On delay constrained multicast capacity of large-scale mobile ad-hoc networks. In *Proc. of IEEE INFOCOM* (2010).