On the Delay and Throughput Gains of Coding in Unreliable Networks

A. Eryilmaz, A. Ozdaglar, M. Médard, E. Ahmed
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA, 02139
Email: {eryilmaz, asuman, medard, ebad}@mit.edu

***Abstract*— In an unreliable packet network setting, we study the performance gains of optimal transmission strategies in the presence and absence of coding capability at the transmitter, where performance is measured in delay and throughput. Although our results apply to a large class of coding strategies including Maximum Distance Separable (MDS) and Digital Fountain codes, we use random network codes in our discussions because these codes have a greater applicability for complex network topologies. To that end, after introducing a key setting in which performance analysis and comparison can be carried out, we provide closed form as well as asymptotic expressions for the delay performance with and without network coding. We show that the network coding capability can lead to arbitrarily better delay performance as opposed to traditional strategies.**

I. Introduction

There has been a growing interest in developing new transmission strategies for efficient use of scarce resources in wireless networks. This is mainly motivated by emerging bandwidth intensive applications such as downloading video or music files, which involves transmission of files to multiple (potentially heterogeneous) receivers. While the standard approach to data transmission builds on the scheduling approach, where information is transmitted to one of multiple receivers as a function of their channel conditions, it has been recognized that broadcasting to multiple receivers using *network coding* may improve performance in such settings. A fundamental question is to understand and quantify the performance gains obtained from network coding in wireless networks.

There has been considerable effort in revealing various gains of network coding. For example, in recent works [3, 20], it has been shown that network coding provides significant buffer savings over traditional methods. Most of the existing research to date has focused on throughput gains obtained from network coding (c.f. [1, 12, 11, 13]). Although these throughput gains may appear to imply gains in delay through Little's law, this is not clearly the case since coding is performed over large blocks and each packet in the block must await the completion of the whole block before it can be decoded. To capture these effects, one must study the system at the packet level instead of using the flow-level formulation of delay (see e.g. [24]). Despite considerable practical interest in the use of network coding in wireless communication systems, gains in delay performance resulting from network coding relative to traditional scheduling have not been analyzed or quantified.

In this paper, we develop a model to study delay performance of network coding and traditional scheduling strategies in unreliable networks. To that end, we consider a scenario where a sequence of incoming files to a transmitter are to be communicated over the time-varying wireless medium to a set of neighboring receivers. This model not only captures the cellular and satellite downlink communications, but also serves as a building block for the operation and analysis of multi-hop wireless networks, as will be discussed in this paper.

We assume that files are broadcast to the receivers in a *rateless* fashion, i.e., the subsequent transmissions do not start before the whole of the current file is received by all the interested receivers. Our goal is threefold. First, we identify the optimal strategies under two transmission modes, namely scheduling and network coding, and quantify and compare the delay performance. Second, we use this model to investigate the sensitivity of the delay gains of network coding to key system parameters such as the number of receivers in the system and the file size. Third, we show how these results can be extended to more general network settings.

Our model involves transmission of (multiple) files from a single transmitter to multiple receivers with varying channel conditions. The varying channel conditions are modeled as stochastic changes in ON/OFF state of the channel. We analyze the model both when Channel Side Information (CSI) about the state of receiver channels is available to the base station and when transmission must be carried without such information.

First part of our analysis focuses on the key scenario of transmitting a single flow, where a flow is a sequence of files generated according to a random process, and destined to the same set of receivers. We consider a dynamic traffic model, in which the files associated with the flow arrive according to a Poisson process. As a measure of performance, we first focus on the mean value of the completion time, which is defined as the time required to transmit all packets of the head-of-line file to all the receivers. For this metric we establish the following results: For the network coding mode, we show that the random linear coding introduced by Ho *et al.*[8] is optimal, in the sense of minimizing the mean completion time, both with and without CSI. This is interesting because it provides simple transmission strategy with no requirement of feedback, but still achieve optimal performance. For the scheduling mode, the presence of CSI affects the optimal strategy. While without the CSI, the optimal scheduling policy is the Round Robin (RR), the characterization of the optimal policy in the presence of CSI necessitates a dynamic programming formulation, which we provide in the paper. Since the computation of the optimal policy using this formulation becomes intractable as the size of the problem increases, we also present an efficient heuristic policy which we use for numerical comparisons. Our numerical analysis shows that network coding leads to a significant improvement in mean completion time with respect to scheduling both with and without CSI.

As a complementary measure of performance of the system, we consider the mean value of the waiting time of an incoming packet, which is defined as the average time between a typical file's arrival and the completion of its service. It known from queueing literature that the mean waiting time is a function of the first and second moments of the completion time. For random linear coding, we provide closed-form expressions for the first and second moments of the completion time. However since these expressions are in terms of infinite-sums, they do not enable us to do sensitivity analysis with respect to system parameters. We therefore provide asymptotic approximations to the first and second moments which highlight the explicit dependence on key parameters. These asymptotic expressions for the moments of mean service time with network coding are new and should be of independent interest in the analysis of coded-networks.

For the RR scheduler, we present bounds on the first and second moments of the waiting time. These results allow us to study asymptotic gains of network coding compared to scheduling and establish a number of sensitivity results. In particular, our analysis shows the delay and throughput gains of network coding compared to scheduling as a function of file size and the number of receivers. Our analysis proves that in the dense network setting where the number of receivers is large, achievable throughput of network coding relative to scheduling scales linearly with the file size, while the mean waiting time of scheduling relative to network coding for the same load scales quadratically with the file size.

In the second part of our analysis, we focus on another canonical scenario where multiple streams are downloaded to a different set of receivers. For this scenario, we present the optimal transmission strategies under both scheduling and coding modes. We establish the following results: we show that a variant of the Longest Connected Queue (LCQ) policy introduced in [21] is the optimal network coding strategy; we prove that coding across sessions (*intersession* network coding) is not favorable for our system; we characterize the optimal scheduling strategies both with and without CSI, and observe significant gains from network coding when CSI is not available. These findings are important in identifying the optimal methods to be employed when multiple flows are to be served by the transmitter.

Our paper differs from the existing work in this area by explicitly modeling delay performance in file downloads and allowing for transmission without CSI. Previous research has instead focused on either optimal scheduling with time-varying channel conditions (see [21, 22]), or on the capacity gains from network coding (see [15, 9, 10, 18, 14, 23]) under various different scenarios. This work builds on the findings of [5], which provided the first quantification of delay gains of network coding by using mean service time as the performance metric. In a more recent independent work [6], Ghaderi *et al.* provided an asymptotic formulation of the mean delay gains by building on the work of Grabner *et al.* ([7]). In this work, we extend these results by considering dynamic arrivals and studying the mean waiting time to provide exact as well as asymptotic expressions for the gains of network coding and scheduling. In another independent work [16], Nguyen *et al.* study the network coding performance in a single-hop broadcast setting in the presence of acknowledgements from the receivers.

The first part of our work is most closely related to [19], where the authors study a multicast scenario with stochastically arriving packets and provide a transform-based analysis of delay for arbitrary coding window sizes. This approach, while providing explicit characterizations of the distributions of the arrival and service processes, does not reveal the relationship between the delay performance and the critical system parameters such as the number of users and the coding window size. Yet, characterization of such a relationship is important in understanding the impact of essential system parameters on performance, and hence in providing valuable insights for the design of efficient systems. In this paper, we exert considerable effort and utilize completely different machinery such as Mellin transforms to obtain an asymptotically accurate formulation of delay with respect to the number of users, channel statistics, and the coding window size. Also, different from [19], we study the scenario of multiple unicast sessions and discuss ways of extending the analysis to the multi-hop network setting.

The rest of the paper is organized as follows: In Section II, the system model is introduced along with the transmission modes of interest and our goals. In Section III, the key scenario of a single flow destined to all the receivers is analyzed in detail. Section IV considers the case of multiple unicast flows. A method to extend the results to more general network settings is suggested in Section V. Finally, we provide a summary and our concluding remarks in Section VI.

II. System model and Goals

In this section, we describe the single-hop setting of one transmitter broadcasting to multiple receivers over independently time-varying channels. This setting not only models the characteristics of cellular or satellite systems, but also serves as the fundamental building block for more general networks. The connection to general topologies will be made explicit in Section V.

a) Single-hop Setting: Consider a single transmitting node and a set $\mathcal{N}$ of receiving nodes that are connected to it over time-varying channels. We assume a time-slotted system to which all the nodes are assumed to be synchronized. The duration of each time slot is selected with respect to the coherence time of the associated system so that channels stay constant within each slot, and vary across time-slots.

A set $\mathcal{F}$ of *flows* generates a sequence of *files* to be multicast to a subset of the receivers. Specifically, Flow $f \in \mathcal{F}$ is demanded by the set $\mathcal{N}_f \subseteq \mathcal{N}$ of receivers[1]. Files associated with each flow arrive at (or are generated by) the transmitter according to a stochastic process. Each file associated with a flow f is composed of K_f *packets*, each of which is a vector of length m over a finite field $\mathbb{F}_d$. We assume that the duration of a time-slot can accommodate a single packet. The files of each flow are accumulated in a separate queue[2] to be transmitted in a First-In-First-Out (FIFO) manner. We assume that transmission of a file starts only after the transmission of the file prior to it in the queue is complete.

The channel between the transmitter and the i^{th} receiver is a randomly varying ON/OFF channel. We let $C_i[t] \in \{0,1\}$ denote the state of user i's channel in slot t. We assume that Receiver-i successfully receives the packet transmitted at slot t if $C_i[t] = 1$, and it cannot receive anything if $C_i[t] = 0$. We will take each $C_i[t]$ to be a Bernoulli random variable with mean p_i that are independent across time and across receivers. The channels of different receivers can in general be asymmetric, i.e., p_i may be different for different $i \in \{1, \cdots, N\}$. However, in parts of the subsequent analysis we will restrict our attention to the symmetric case of $p_i = p$ for all i in order to have tractable formulations. The system model is depicted in Figure 1.

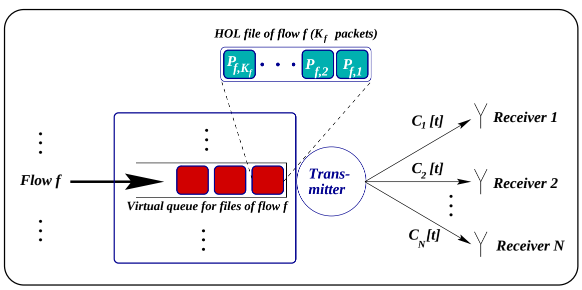

Fig. 1. System model

b) Availability of Channel Side Information (CSI): We distinguish between two cases regarding the availability of CSI at the transmitter. We say that CSI is available when the vector[3] of channel states $\mathbf{C}[t] \triangleq (C_1[t], \cdots, C_N[t])$ is known by the transmitter at the beginning of slot t so that transmissions can be decided with the perfect knowledge of which receivers will get them. Such an assumption requires extra overhead for estimation and feedback operations, and may be impractical especially when the number of receivers is large or the channel variations are too fast to accommodate the feedback delay. We study this scenario under the assumptions of perfect and instantaneous feedback with negligible overhead as a limiting idealistic case. The outcome of this study will allow us to identify the strengths of weaknesses of different strategies even when CSI is available.

The more realistic scenario of NO-CSI refers to the case when no channel quality information is available to the transmitter at the outset of transmission. Thus, the decision as to what to transmit must be made blindly. In this extreme, we assume that feedback is very costly, and hence must be minimized. This is a reasonable assumption when the number of receivers is large. Thus, instead of intermediate scenarios such as ARQ-type schemes, we focus on the case where the receivers send acknowledgement (ACK) packets only when they receive the whole file. Thus, we assume a file-based ACK scheme, rather than the significantly more costly packet-based ACK scheme.

c) Transmission Strategies: The strategy employed by the transmitter to broadcast the head-of-the-line file to the receivers has a critical effect on the service time distribution of the file completion. We focus on two modes of transmission in this paper, namely *scheduling* and *coding*. Before we define these two modes, we introduce some notation. Since the files are transmitted in a FIFO order, we can focus on the head-of-line (HOL) file of flow f, which is composed of K_f packets. Then, Packet-k of the HOL file of flow f is referred to as $\mathbf{P}_{f,k}$, which is a vector of length m over a finite field $\mathbb{F}_d$. Finally, let $\mathbf{P}[t]$ denote the packet chosen for transmission in slot t.

Definition 1 (Scheduling). Scheduling *refers to the mode of transmission where in any given slot, the transmitter must pick a single packet from the HOL file to transmit. Specifically, we have* $\mathbf{P}[t] \in \{\mathbf{P}_{f,k} | f \in \mathcal{F},\ k = 1, \cdots, K_f\}$.

Definition 2 ((Network) Coding). (Network) Coding *refers to the mode of transmission where in every slot, say t, any linear combination of the packets belonging to the HOL file can be transmitted. Specifically, we have*

$$\mathbf{P}[t] = \sum_{f \in \mathcal{F}} \sum_{k=1}^{K_f} a_{f,k}[t] \mathbf{P}_{k,f},$$

where $a_{f,k}[t] \in \mathbb{F}_d$ for each $f \in \mathcal{F}$ and $k \in \{1, \cdots, K_f\}$. The transmitter chooses the coefficients $\{a_{f,k}[t]\}$ for each t.

d) Goals: For the above model, we are interested in

- identifying the *optimal transmission strategies* under scheduling and coding transmission modes, and under the assumption of CSI and NO-CSI, where the *optimal strategy* is the one which minimizes the mean service time;
- providing an analytical expression of the mean waiting time (including queueing delay and service time) for the incoming packets under the optimal transmission schemes;
- understanding the asymptotic effect of the number of users and the file sizes on the mean waiting time;
- providing methods for extending the single-hop setting to multiple-hop networks with general topologies.

We will address each of these goals in the subsequent analysis.

The rest of the paper is organized as follows. In Section III, we focus on the broadcast scenario where the incoming files are to be transmitted to all the receivers. After characterizing the optimal transmission strategies, we provide explicit as well as asymptotic expressions for their performance and show the significant gains of coding with respect to scheduling. In Section IV, we consider the scenario of multiple unicast

[1]We will use F, N and N_f to denote the cardinalities of the sets $\mathcal{F}, \mathcal{N}$ and $\mathcal{N}_f$, respectively.

[2]This queue need not be a physically separate buffer, but a virtual one where files of different flows are accounted for separately.

[3]We will consistently use **boldface** letters to denote vectors.

flows, where each flow is destined to a separate receiver, and characterize the optimal policies in the two transmission modes. In Section V, we provide a method whereby the results and analysis of the single hop setting can be extended to multi-hop networks. Finally, we provide our concluding remarks in Section VI.

III. Broadcasting a Single Flow

In this section, we focus on the key scenario of the transmitter broadcasting the incoming files of a single flow to all the receivers, i.e., we set $F = 1$, and $\mathcal{N}_f = \mathcal{N}$ in our model. This scenario allows us to isolate the delay analysis from issues of scheduling transmissions across flows, and allows for tractable analysis. Since there is only one flow in the system, we will drop the subscript f in our notation throughout this section, and denote Packet-k of the HOL file of the flow as $\mathbf{P}_k$, and the size of the file as K.

We assume that files of the flow arrive according to a Poisson process[4] of rate λ. The Poisson assumption allows us to view the whole system as an $M/G/1$ queue, where the service time distribution is a function of the transmission strategy being employed at the transmitter. Let $Z(N, K)$ denote the time required to transmit all the packets of the HOL file to all the receivers under a given transmission strategy, and (N, K) parameters. We refer to $Z(N, K)$ as the *completion* (or *service*) *time* of a file download. The mean waiting time $\overline{W}(\lambda, N, K)$ of an incoming file is given by the *Pollaczek-Khinchin* formula ([2]):

$$\overline{W}(\lambda, N, K) = \frac{\lambda \mathbb{E}[Z(N, K)^2]}{2(1 - \lambda \mathbb{E}[Z(N, K)])}, \quad (1)$$

It is seen from (1) that the mean waiting time is a function of the first and second moments of the HOL file completion time. In Section III-A, we identify the optimal transmission strategies under the scheduling and coding modes of operation, where optimality is in terms of minimizing the mean completion time. Then, in Section III-B, we provide closed form as well as asymptotic expressions for the first and second moments of the completion time under the identified optimal strategies. Numerical as well as asymptotic performance comparison of the two transmission strategies will be provided in Section III-C. This investigation will reveal the delay gains of network coding with respect to traditional scheduling strategies in unreliable wireless systems.

A. Optimal Transmission Strategies

The aim of this section is to identify those coding and scheduling strategies that lead to minimum mean completion time of the HOL file, both in the presence and lack of CSI. It can be seen by looking at (1) that the mean service time of a policy is the key factor in determining the maximum arrival rate λ that the policy can support with a finite delay. This is our motivation for focusing on minimizing this performance criterion. Next, we focus on the coding and scheduling cases separately.

[4]For other arrival processes, various bounds such as Kingman's bound can be used to characterize the system delay.

1) Optimal Coding Strategy with and without CSI: It has been shown in the literature that linear coding is sufficient to achieve the maximum achievable rate for a single multicast session in general networks [12]. Noticing that the broadcast scenario is a special instance of a multicast transmission, we focus on linear coding strategies where the transmitted packet in slot t is given by $\mathbf{P}[t] = \sum_{k=1}^{K} a_k[t]\mathbf{P}_k$, with $a_k[t] \in \mathbb{F}_d$ for each $k \in \{1, \cdots, K\}$.

Proposition 1. *The following randomized strategy is asymptotically optimal as the field size d tends to infinity: The transmitter performs the following operation for the HOL file*

RANDOMIZED BROADCAST CODING (RBC):
While (File is incomplete)
Pick $a_k[t]$ uniformly at random from $\mathbb{F}_d$ for each k;
Transmit $\mathbf{P}[t] = \sum_{k=1}^{K} a_k[t]\mathbf{P}_k$;
$t \leftarrow t + 1$;

Each receiver keeps the incoming packets that it could receive and then decodes all the packets $\{\mathbf{P}_k\}_{\{k=1,\cdots,K\}}$ as soon as K linearly independent combinations of the packets are collected (c.f. [8] and references therein). Finally, each receiver that successfully recovers the HOL file sends an acknowledgement to the transmitter.

Proof: The expected number of slots before K linearly independent combinations can be collected with Randomized Broadcast Coding (RBC) is given by $\sum_{k=1}^{K}[1 - (1/d)^k]^{-1}$. This expression can be upper-bounded by $Kd/(d-1)$, which in turn can be made close to K even with reasonably low values of d. Thus, for a large enough field size d, it is sufficient for each receiver to be active approximately K slots before it can decode the whole file. Notice that it is impossible to send the file with less than K transmissions since at most one packet can be successfully transmitted in one transmission, and so RBC asymptotically (in d) achieves the best possible performance over all strategies. ■

Another important issue is the overhead related with this mode of transmission. Coding requires $\lceil K \log_2 d \rceil$ bits of overhead to contain the coefficients of the associated linear combination, whereas the packet size is $\lceil m \log_2 d \rceil$ bits. Thus, for $m >> K$, the overhead is negligible. Henceforth, we will consider this scenario, and ignore the overhead.

Notice that RBC is not only easy to implement, but also requires no knowledge of the channel state vector, and asymptotically achieves the minimum mean completion time over all policies. We will see in Section III-A.2 that the optimal scheduling policy is much more difficult to characterize, even for symmetric channel conditions.

2) Scheduling Mode: In this mode, unlike in the coding mode, the presence or lack of CSI affects the performance. Hence, these two cases will be studied separately.

a) Scheduling without CSI: In view of the assumptions that the transmitter receives feedback from each receiver only at the completion of the whole file and that the channels are symmetric, we can see that all packets in the HOL file have equal priorities. Therefore, we have the following result.

Proposition 2. *Assuming NO-CSI and independent and iden-*

tically distributed (i.i.d.) channels across time-slots and users, the optimal scheduling policy is Round Robin *(RR), where Packet-k is transmitted in time-slots* $(mK + k)$ *for* $m = 0, 1, \cdots$ *until all the receivers get the file.*

Proof: This follows from the perfectly symmetric conditions assumed under this scenario. ■

b) Scheduling with CSI: Before we characterize the optimal scheduling rule with CSI, we demonstrate the suboptimality of scheduling compared to coding with the following key example.

Example 1: Consider the case of $K = 3$ and $N = 3$, i.e. three packets are to be broadcast to three receivers. Consider the channel realizations $\mathbf{C}[1] = (0, 1, 1), \mathbf{C}[2] = (1, 0, 1), \mathbf{C}[3] = (1, 1, 0)$, and $\mathbf{C}[4] = (1, 1, 1)$. Thus, in the first four slots, each receiver can hear the transmission three times. The optimal scheduling rule would transmit $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ in the first three slots, leaving Receiver-i in demand for Packet-i in the fourth slot. Clearly, no scheduling rule can ever complete the file download at all three receivers in the fourth slot. With coding, on the other hand, the following transmissions will complete the transmissions: $(\mathbf{P}_1+\mathbf{P}_2), (\mathbf{P}_2+\mathbf{P}_3), (\mathbf{P}_1+\mathbf{P}_3), (\mathbf{P}_1+\mathbf{P}_2+\mathbf{P}_3)$ (see Table I). It is not difficult to see that coding will never require more slots than is necessary for scheduling for all other realizations. Hence, we achieve strictly better completion times with coding. ◇

	$t=1$	$t=2$	$t=3$	$t=4$
R_1	$-$	$\mathbf{P}_2 \vert (\mathbf{P}_2+\mathbf{P}_3)$	$\mathbf{P}_3 \vert (\mathbf{P}_1+\mathbf{P}_3)$	$? \vert (\mathbf{P}_1+\mathbf{P}_2+\mathbf{P}_3)$
R_2	$\mathbf{P}_1 \vert (\mathbf{P}_1+\mathbf{P}_2)$	$-$	$\mathbf{P}_3 \vert (\mathbf{P}_1+\mathbf{P}_3)$	$? \vert (\mathbf{P}_1+\mathbf{P}_2+\mathbf{P}_3)$
R_3	$\mathbf{P}_1 \vert (\mathbf{P}_1+\mathbf{P}_2)$	$\mathbf{P}_2 \vert (\mathbf{P}_2+\mathbf{P}_3)$	$-$	$? \vert (\mathbf{P}_1+\mathbf{P}_2+\mathbf{P}_3)$

TABLE I. Demonstration of *Example 1*: R_i corresponds to Receiver-i, '–' denotes OFF channel states, and the entry $\mathbf{a}|\mathbf{b}$ gives the optimal transmissions with scheduling|coding, respectively. With scheduling, no choice of $\{\mathbf{P}_i\}$ in slot 4 can complete the file at all the receivers for the given channel realization.

OPTIMAL SCHEDULING RULE WITH CSI: We use Dynamic Programming to find the characterization of the optimal scheduling policy. Given $\mathbf{C}[t]$, the scheduler can choose any one of the packets $\{\mathbf{P}_1, \cdots, \mathbf{P}_K\}$ for transmission. A little thought reveals the need for memory at the transmitter about the history of receptions of each receiver. For this purpose, we define $M_{i,k}[t]$ to be the memory bit associated with Packet-k and Receiver-i. In particular, $M_{i,k}[t] = 1$ (or 0) implies that Receiver-i has not received (or has received) Packet-k in the slots $1, \cdots, t-1$. Moreover, we will use $\mathbb{M}[t]$ to denote the matrix of memory bits $[M_{i,k}[t]]_{i=1,\cdots,N}^{k=1,\cdots,K}$.

We let Π denote the set of feasible stationary policies that can be implemented by the transmitter. Each policy $\pi \in \Pi$ defines a mapping from the pair $(\mathbb{M}[t], \mathbf{C}[t])$ to the set $\{1, \cdots, K\}$ describing the packet to be sent at time t. Note that the policy is stationary in the sense that it is only a function of the matrix and channel conditions at the time. The i.i.d. nature of the arrivals and departures imply that this is the optimal policy among all policies, including those that are time dependent.

To characterize the optimal policy we let $J^{\pi}(\mathbb{M}, \mathbf{C}) = \mathbb{E}\left[\# \text{ slots to reach } \theta \text{ with policy } \pi \mid \mathbb{M}[0] = \mathbb{M},\ \mathbf{C}[0] = \mathbf{C}\right]$, where θ denotes the zero matrix. Then, $J^{\star}(\mathbb{M}, \mathbf{C}) \triangleq \min_{\pi \in \Pi} J^{\pi}(\mathbb{M}, \mathbf{C})$ is the minimum completion time of the optimal algorithm if it starts from $\mathbb{M}$ and the first channel is $\mathbf{C}$. Also, $\pi^{\star}(\mathbb{M}, \mathbf{C}) \triangleq \arg\min_{\pi \in \Pi} J^{\pi}(\mathbb{M}, \mathbf{C})$ gives the optimal policy.

Observe that once we solve $J^{\star}(\mathbb{M}, \mathbf{C})$ for all $\mathbf{C}$, we can compute $J^{\star}(\mathbb{M}) \triangleq \mathbb{E}_{\mathbf{C}}\left[J^{\star}(\mathbb{M}, \mathbf{C})\right]$, where the expectation is over the channel realizations. Thus, $J^{\star}(\mathbb{M})$ denotes the mean completion time of the optimal algorithm starting from $\mathbb{M}$. Hence, we are interested in $J^{\star}([1]_{N\times K})$ where $[a]_{N\times K}$ denotes the all a matrix of dimensions $N \times K$.

Before we write the recursion for $J^{\star}(\mathbb{M}, \mathbf{C})$, let us define the function $f(\cdot)$ where $\hat{\mathbb{M}} = f(\mathbb{M}, \mathbf{C}, k)$ implies that

$$\begin{aligned} \hat{M}_{i,k} &= M_{i,k} - M_{i,k} C_i & \forall i \in \{1, \cdots, N\}, \\ \hat{M}_{i,j} &= M_{i,j} & \forall i \in \{1, \cdots, N\}, j \neq k. \end{aligned}$$

This function describes the next state of the memory matrix given that Packet-k is served and the channel matrix is $\mathbf{C}$ in the current slot. Then, we can write the following recursion:

$$J^{\star}(\mathbb{M}, \mathbf{C}) = \arg\min_{k \in \{1, \cdots, K\}} \left\{ J^{\star}(f(\mathbb{M}, \mathbf{C}, k)) + \mathbf{1}_{\{\mathbb{M} \neq \theta\}} \right\},$$

where $\mathbf{1}_{\{A\}}$ is the indicator function of the event A.

The monotone nature of the $f(\cdot)$ function enables us to compute $J^{\star}(\mathbb{M}, \mathbf{C})$ and $\pi^{\star}(\mathbb{M}, \mathbf{C})$ recursively starting from the base state $J^{\star}(\theta) = 0$ (c.f. [4]). This DP formulation characterizes the optimal policy and its performance, and can be computed starting from a 1×1 matrix and increasing N and K successively.

However, as N and K grows, the necessary number of operations required to find the optimal strategy grows exponentially and quickly becomes impossible to handle. Thus, we propose an efficient heuristic policy below and simulate its performance for comparison.

HEURISTIC POLICY: We have observed in the above discussions that the optimal scheduling rule has a complicated structure. Yet, it is possible to find practical scheduling algorithms that performs close to the optimal. Here, we describe a heuristic policy that achieves near optimal performance based on numerical comparisons.

At any given time slot t, let us denote the set of nodes with an ON channel (also called the set of *active receivers*) by $\mathcal{A}[t] \triangleq \{i \in \{1, \cdots, N\} : C_i[t] = 1\}$. Under the symmetric conditions that we assumed, the packet that would provide the most *benefit* should intuitively be transmitted over the channel. We propose that the benefit of a packet be measured in the number of nodes in $\mathcal{A}[t]$ that has not yet received that packet. The underlying idea is to transfer the maximum number of useful packets over the channel at any given time. These remarks point to the heuristic algorithm given next.

HEURISTIC BROADCAST SCHEDULING (HBS):

```
If (t = 1)
    M_{i,k}[t] ← 1 for all k ∈ {1, ··· , K}, i ∈ {1, ··· , N};
While ( Σ_{k=1}^{K} Σ_{i=1}^{N} M_{i,k}[t] > 0 )
    K[t] ≜ {k ∈ {1, ··· , K} : ∃i ∈ A[t] with M_{i,k}[t] = 1};
    If (K[t] ≠ ∅)
        T[t] ≜ arg max_{k∈K[t]} Σ_{i∈A[t]} M_{i,k}[t];
        Pick a k* ∈ T[t];
        M_{i,k*}[t] ← 0 for all i ∈ A[t];
        Transmit Packet-k* over the channel at slot t;
    t ← t + 1;
```

In the algorithm, each packet in $\mathcal{K}[t]$ has at least one receiver with an ON channel in slot t which demands that packet. Clearly, those packets that are not in $\mathcal{K}[t]$ should not be chosen for transmission. If $\mathcal{K}[t] \neq \emptyset$, then we define $\mathcal{T}[t]$ to be the set of packets in $\mathcal{K}[t]$ that yield the most benefit in slot t. Then, a packet from $\mathcal{T}[t]$ is picked for transmission in slot t. In our simulations, we considered a random picking of one of the packets in $\mathcal{T}[t]$. However, the performance can be slightly improved by using more sophisticated methods. For example, for $N = 2$, the packet picked from $\mathcal{T}[t]$ may be chosen amongst those packets that has already been received by the OFF receiver. Then, every time a receiver is ON, it will receive a useful packet until all its packets are complete. Thus, this algorithm gives the optimal policy for $N = 2$. The generalization of the picking method to general K under asymmetric channel conditions is complicated and requires increasing memory to operate. On the other hand, the complexity of HBS at each iteration of the loop is $O(KN)$ and requires no extra memory, and hence it is relatively easy to implement.

B. Service Time Distributions

The goal of this section is to provide analytical and asymptotic performance expressions for mean waiting time under the optimal coding and scheduling transmission strategies identified in Section III-A. The exact analytical expressions provided here are in terms of infinite sums, and therefore do not yield much insight about the impact of system parameters on the performance. Here, we also derive asymptotic expressions to provide a sensitivity analysis with respect to key system parameters. We focus on the more realistic scenario of NO-CSI throughout this section. Our arguments are based on deriving expressions for the first and second moments of the completion time under coding and scheduling, and then using them in (1) to get the mean waiting time performances.

1) Performance Analysis of RBC: Let us define the random variable Y_i^{RBC} as the number of slots before Receiver-i's channel is ON K times, for $i = 1, \cdots, N$. Then, the completion time under RBC for a given N and K, denoted by $Z^{RBC}(N,K)$, satisfies

$$Z^{RBC}(N,K) = \max_{i\in\{1,\cdot,N\}} Y_i^{RBC}, \tag{2}$$

which is the maximum of N Pascal variables of order K. We will use m_1^{RBC} and m_2^{RBC} denote the first and second moments of $Z^{RBC}(N,K)$, respectively. Through algebraic manipulations, we can derive closed-form expressions for these moments. As an example, the first moment is given by

$$m_1^{RBC} = K + \sum_{t=K}^{\infty}\left[1 - \prod_{i=1}^{N}\left(\sum_{\tau=K}^{t}\begin{pmatrix}\tau-1\\K-1\end{pmatrix} q_i^{(\tau-K)} p_i^K\right)\right],$$

where $\begin{pmatrix}n\\m\end{pmatrix}$ gives the number of size m combinations of n elements, and $q_i \triangleq (1-p_i)$. Similarly, a combinatorial expression can be given for the second moment. For simplicity of exposition, we provide the second moment for the symmetric channel conditions:

$$\begin{aligned} m_2^{RBC} &= \sum_{i=1}^{\infty} i^2 \left[\left(\sum_{\tau=K}^{i}\begin{pmatrix}\tau-1\\K-1\end{pmatrix} q^{(\tau-K)} p^K\right)^N \right. \\ &\quad \left. - \left(\sum_{\tau=K}^{i-1}\begin{pmatrix}\tau-1\\K-1\end{pmatrix} q^{(\tau-K)} p^K\right)^N\right]. \end{aligned}$$

Although the exact expressions provided above can be used for numerical comparison, the expressions can be simplified by focusing on the asymptotic regime for the symmetric case. Such an asymptotic study has the added advantage of revealing the gains of coding versus scheduling as a function of relevant system parameters. The asymptotic formulations are especially useful to understand the gains in dense networks, where an increasing number of transceivers are used within a fixed geographic area.

The next proposition, proved in [7], will be used in our subsequent analysis. It provides an expression for an infinite sum that is directly related to m_1^{RBC} as will be noted in Proposition 4.

Proposition 3 ([7]). *Let $g(r) = \beta r^\alpha$ and let $\mathrm{lq}(\cdot)$ be a shorthand for $\log_{\frac{1}{q}}(\cdot)$, then*

$$\begin{aligned} &\sum_{r\geq 0}\left(1-(1-g(r)q^r)^N\right) \\ &\quad = \mathrm{lq}\,N + \alpha\,\mathrm{lq}\,\mathrm{lq}\,N + \mathrm{lq}\,\beta + \frac{1}{2} - \frac{\gamma}{\log q} \\ &\qquad + h(\mathrm{lq}\,N + \alpha\,\mathrm{lq}\,\mathrm{lq}\,N + \mathrm{lq}\,\beta) + o(1), \end{aligned}$$

where γ is the Euler-Mascheroni constant (approximately equal to 0.5772), and $h(\cdot)$ is a periodic C^∞-function of period 1 and mean value 0, whose Fourier coefficients are $\hat{h}(k) = \frac{1}{\log q}\Gamma\left(\frac{2ik\pi}{log q}\right)$, for $k \in \mathbb{Z}_+$. □

The next proposition provides asymptotic expressions for m_1^{RBC} and m_2^{RBC} under symmetric conditions as a function of N and K.

Proposition 4. *Assume symmetric channel conditions, i.e., $p_i = p$ for all $i \in \{1, \cdots, N\}$, and let $\mathrm{lq}(\cdot)$ be a shorthand for $\log_{\frac{1}{q}}(\cdot)$. Then, we have*

$$\begin{aligned} m_1^{RBC} &= \mathrm{lq}\,T + \frac{1}{2} - \frac{\gamma}{\log q} + h(\mathrm{lq}\,T) + o(1), \\ m_2^{RBC} &= \mathrm{lq}^2\,T + \mathrm{lq}\,T(1 + 2\gamma + 2g_1(\mathrm{lq}\,T)) + \frac{2}{3} \\ &\quad - \frac{\gamma}{\log q} - \frac{(\gamma^2 + (\pi^2/6))}{\log^2 q} + O((K-1)\,\mathrm{lq}\,\mathrm{lq}\,N) \\ &\quad + h(\mathrm{lq}\,T) + g_2(\mathrm{lq}\,T) + o(1), \end{aligned}$$

where $T = N\left(\frac{p}{q}\right)^{K-1}\frac{\mathrm{lq}^{(K-1)}N}{(K-1)!}$, *and* $h(\cdot)$ *is the periodic function of Proposition 3, and* $g_1(\cdot)$, *and* $g_2(\cdot)$ *are two periodic* C^∞*-functions of period* 1 *and mean* 0.

Proof: We outline the proof of m_1^{RBC} which is due to [7]. In the sequel, let us use Z and Y_i as shorthands for $Z^{RBC}(N,K)$ and $Y_i^{RBC}(N,C)$ for convenience. Also, we use $F_\star(\cdot)$ generically to denote the cumulative distribution of the random variable $\star$ in the subscript.

Since $\{Y_i\}$ are i.i.d. Pascal random variables of order K, we have

$$F_Y^c(m) := 1 - F_Y(m) = \sum_{k=0}^{K-1}\binom{m}{k}p^k q^{m-k} =: q^m g(m),$$

where

$$\begin{aligned} g(m) &:= \sum_{k=0}^{K-1}\binom{m}{k}\left(\frac{p}{q}\right)^k \\ &\sim \frac{m^{K-1}}{(K-1)!}\left(\frac{p}{q}\right)^{K-1} =: \beta m^\alpha, \end{aligned} \tag{3}$$

with $\beta := \left(\frac{p}{q}\right)^{K-1}\frac{1}{(K-1)!}$, and $\alpha := (K-1)$. The last approximation is accurate for $m \gg K$ since the last term of the sum dominates, and $\binom{n}{k} \sim \frac{n^k}{k!}$ for $n \gg k$. Then, we can write

$$\begin{aligned} m_1^{RBC} &= \sum_{m\geq 0}\left(1 - F_{max_i Y_i}(m)\right) \\ &= \sum_{m\geq 0}\left(1 - (1-g(m)q^m)^N\right). \end{aligned}$$

Notice that the final expression is in the form of Proposition 3. The proof is complete when we apply the result of Proposition 3 with $g(m) = \beta m^\alpha$ as defined above.

Next, we prove the expression for m_2^{RBC}. Note that

$$\begin{aligned} m_2^{RBC} &= \mathbb{E}[Z^2] \\ &= \sum_{k\geq 0}\left(1 - F_{Z^2}(k)\right) \\ &= \sum_{k\geq 0}\left(1 - F_{\max_i Y_i}(\lfloor\sqrt{k}\rfloor)\right) \\ &= \sum_{k\geq 0}\left(1 - \left[F_Y(\lfloor\sqrt{k}\rfloor)\right]^N\right) \\ &= \sum_{k\geq 0}\left(1 - \left(1 - g(\lfloor\sqrt{k}\rfloor)q^k\right)^N\right), \end{aligned} \tag{4}$$

where $\lfloor x\rfloor$ is the largest integer that is less than or equal to x. Note that following the arguments in (4), we can approximate (3) by replacing $g(m)$ with βm^α as argued above. In order to simplify the $\lfloor\sqrt{\cdot}\rfloor$ in (4), we make a change of variable, and write

$$m_2^{RBC} = \sum_{r\geq 0}(2r+1)\left(1 - (1-g(r)q^r)^N\right). \tag{5}$$

We split the sum in (5) into two as

$$m_2^{RBC} = E_1 + 2E_2, \tag{6}$$

where

$$\begin{aligned} E_1 &\triangleq \sum_{r\geq 0}\left(1 - (1-g(r)q^r)^N\right), \\ E_2 &\triangleq \sum_{r\geq 0} r\left(1 - (1-g(r)q^r)^N\right) \end{aligned}$$

Notice that $E_1 = m_1^{RBC}$, which is already studied above. Next, we derive a similar expression for E_2. To that end, we define

$$\begin{aligned} \tilde{E}_2 &:= \sum_{r\geq 0} r\left(1 - \exp(-N\beta r^\alpha q^r)\right) \\ \hat{E}_2 &:= \sum_{r\geq 0} r\left(1 - \exp(-Tq^r)\right), \end{aligned}$$

where $T := N\beta\,\mathrm{lq}^\alpha N = N\left(\frac{p}{q}\right)^{K-1}\frac{\mathrm{lq}^{(K-1)}N}{(K-1)!}$. Then, we can write

$$E_2 = \underbrace{(E_2 - \tilde{E}_2)}_{\tilde{\Delta}} + \underbrace{(\tilde{E}_2 - \hat{E}_2)}_{\hat{\Delta}} + \hat{E}_2. \tag{7}$$

We first derive an asymptotic expression for $\hat{E}_2$ as a function of T. We then show that $\tilde{\Delta}$ and $\hat{\Delta}$ lead to negligible terms (asymptotically in terms of N and K), and $\hat{E}_2$ dominates. Our derivation for $\hat{E}_2$ is based on taking its Mellin Transform, and then using Mellin inversion to find an explicit expression for its asymptotic form (see for example [17]). The Mellin transform, $\hat{E}_2^\star(s)$ of $\hat{E}_2(T)$ is given by

$$\hat{E}_2^\star(s) = \int_0^\infty \hat{E}_2(T)T^{s-1}dT = \Gamma(s)\frac{q^s}{(q^s-1)^2},$$

for $\mathbb{R}(s) \in (-1,0)$, which uses the fact that

$$\int_0^\infty (1-\exp(-T))T^{s-1}dT = -\Gamma(s), \quad \text{for } \mathbb{R}(s) \in (-1,0).$$

Mellin inversion yields

$$\hat{E}_2(T) = \frac{1}{2\pi i}\int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} -\Gamma(s)\frac{q^s}{(q^s-1)^2}T^{s-1}ds.$$

Shifting the line of integration to the right gives the asymptotic behavior of $\hat{E}_2(T)$ for $T \to \infty$:

$$\begin{aligned} \hat{E}_2(T) &= \frac{\log^2 q + 6\log^2 T - 12\gamma\log(T) - \pi^2 - 6\gamma^2}{12\log^2 q} \\ &\quad + \sum_{l\in\mathbb{Z}\setminus\{0\}} Res_{s=\frac{2\pi i}{\log q}}(-\Gamma(s))\frac{q^s}{(q^s-1)^2}T^{-s} \\ &\quad + \frac{1}{2\pi i}\int_{M-i\infty}^{M+i\infty} -\Gamma(s)\frac{q^s}{(q^s-1)^2}T^s ds, \end{aligned}$$

for any $M > 0$. The remaining integral is $O(T^{-M})$ for any $M > 0$, and the sum of residues gives $g_1(\mathrm{lq}\,T)\,\mathrm{lq}\,T + g_2(\mathrm{lq}\,T)$ with two periodic C^∞-functions of period 1 and mean value 0. Using these results together with our $\mathrm{lq}(\cdot)$ notation, we can re-write the expression for $\hat{E}_2(T)$ as

$$\begin{aligned} \hat{E}_2(T) &= \frac{\mathrm{lq}^2\,T}{2} + (\gamma + g_1(\mathrm{lq}\,T))\,\mathrm{lq}\,T + \frac{1}{12} \\ &\quad - \frac{(6\gamma^2+\pi^2)}{12\log^2 q} + g_2(\mathrm{lq}\,T) + o(1). \end{aligned} \tag{8}$$

Next, we study $\tilde{\Delta}$ introduced in (7). We start by dividing the sum into two parts, while replacing $g(r)$ by its approximation βr^{α}, as follows:

$$\tilde{\Delta} = \sum_{r \le \mathrm{lq}\, N} r\left[\exp(-N\beta r^{\alpha} q^{r}) - (1-\beta r^{\alpha} q^{r})^{N}\right] \quad (9)$$
$$+ \sum_{r > \mathrm{lq}\, N} r\left[\exp(-N\beta r^{\alpha} q^{r}) - (1-\beta r^{\alpha} q^{r})^{N}\right] \quad (10)$$

In the range when $r \le \mathrm{lq}\, N$, we have $q^{r} \ge 1/N$. Therefore, we get

$$(9) \le \sum_{r \le \mathrm{lq}\, N} r\left[\exp(-\beta r^{\alpha}) - (1-\frac{\beta r^{\alpha}}{N})^{N}\right]$$
$$= \sum_{r \le \mathrm{lq}\, N} r\exp(-\beta r^{\alpha})\left[1-\exp\left(-\beta r^{\alpha}\left(\frac{1}{2N}+o(\frac{1}{N})\right)\right)\right],$$

which follows from the fact that $(1-\frac{x}{N})^{N} = \exp(-x(1+\frac{1}{2N}+o(\frac{1}{N})))$. Next, using the approximation $(1-e^{-y}) \approx y$ for small y, which holds for large N, we can further simplify the previous sum to

$$\sum_{r \le \mathrm{lq}\, N} \beta r^{(\alpha+1)} \exp(-\beta r^{\alpha})\left(\frac{1}{2N}+o(\frac{1}{N})\right)$$
$$\le \beta\, \mathrm{lq}^{(\alpha+2)} N\left(\frac{1}{2N}+o(\frac{1}{N})\right),$$

which proves that as N tends to infinity, (9) tends to zero.

To find a bound on (10) we use the Taylor expansions for e^{xN} and $(1-x)^{N}$ to obtain

$$e^{xN} - (1-x)^{N} = O(Nx^{2}).$$

Substituting $x = \beta r^{\alpha}$ in (10) yields

$$(10) = O\left(N\beta^{2}\sum_{r > \mathrm{lq}\, N} r^{2\alpha+1} q^{2r}\right)$$
$$= O\left(\frac{(\mathrm{lq}\, N)^{m}}{N}\right),$$

for some positive integer m that can be computed using the poly-logarithmic function based on α. This shows that $\tilde{\Delta}$ is negligible asymptotically as $N \to \infty$.

Next, we focus on $\hat{\Delta}$ that was introduced in (7). We begin by splitting the sum as

$$\hat{\Delta} = \sum_{r \le \mathrm{lq}\, N} r e^{-N\beta r^{\alpha} q^{r}}\left(e^{-(T-N\beta r^{\alpha})q^{r}} - 1\right) \quad (11)$$
$$+ \sum_{r > \mathrm{lq}\, N} r e^{-Tq^{r}}\left(1-e^{-(N\beta r^{\alpha}-T)q^{r}}\right), \quad (12)$$

where $T = N\beta\, \mathrm{lq}^{\alpha} N$. We study these sums separately. Note that $(11) \le 0$, since in this range $r \le \mathrm{lq}\, N$. Next, we focus on (12): Calculus shows that the summand of (12) is positive and increasing in the range where $r \in (\mathrm{lq}\, N, \mathrm{lq}\, N + \alpha\, \mathrm{lq}\,\mathrm{lq}\, N]$, and is upper-bounded by $\alpha\, \mathrm{lq}\,\mathrm{lq}\, N/\mathrm{lq}^{\alpha} N$. Therefore we have the summation in this region which we further split into two as follows: for $r \in (\mathrm{lq}\, N, \mathrm{lq}\, N + \alpha\, \mathrm{lq}\,\mathrm{lq}\, N]$, we have

$$\sum_{\mathrm{lq}\, N < r \le \mathrm{lq}\, N + \alpha\, \mathrm{lq}\,\mathrm{lq}\, N} r e^{-N\beta\, \mathrm{lq}^{\alpha} N q^{r}}\left(1-e^{-N\beta(r^{\alpha}-\mathrm{lq}^{\alpha} N)q^{r}}\right)$$
$$\le O\left(\frac{(\alpha\, \mathrm{lq}\,\mathrm{lq}\, N)^{2}}{\mathrm{lq}^{\alpha} N}\right),$$

which tends to 0 as N tends to infinity. Next we focus on the range when $r > \mathrm{lq}\, N + \alpha\, \mathrm{lq}\,\mathrm{lq}\, N$. For notational convenience, we let $x := r - \mathrm{lq}\, N$, and rewrite the sum in this range as

$$\sum_{x > \alpha\, \mathrm{lq}\,\mathrm{lq}\, N}\left[(x+\mathrm{lq}\, N)e^{-\beta\, \mathrm{lq}^{\alpha} N q^{x}} \times \left(1-e^{-\beta((x+\mathrm{lq}\, N)^{\alpha}-\mathrm{lq}^{\alpha} N)q^{x}}\right)\right]$$
$$\le \sum_{x > \alpha\, \mathrm{lq}\,\mathrm{lq}\, N}(x+\mathrm{lq}\, N)\left(1-e^{-\beta((x+\mathrm{lq}\, N)^{\alpha}-\mathrm{lq}^{\alpha} N)q^{x}}\right)$$
$$\overset{(a)}{\approx} \sum_{x > \alpha\, \mathrm{lq}\,\mathrm{lq}\, N}(x+\mathrm{lq}\, N)\,\mathrm{lq}^{\alpha} N\left((1+\frac{x}{\mathrm{lq}\, N})^{\alpha}-1\right)$$
$$\overset{(b)}{=} \beta\sum_{x > \alpha\, \mathrm{lq}\,\mathrm{lq}\, N}(x+\mathrm{lq}\, N)\,\mathrm{lq}^{\alpha} N\sum_{m=1}^{\alpha}\binom{\alpha}{m}\frac{x^{m}}{\mathrm{lq}^{m} N}q^{x}$$
$$= \beta\sum_{m=1}^{\alpha}\binom{\alpha}{m}\mathrm{lq}^{(\alpha-m)} N\sum_{x > \alpha\, \mathrm{lq}\,\mathrm{lq}\, N}(x^{m+1}+\mathrm{lq}\, N x^{m})q^{x}$$
$$\overset{(c)}{\le} O\left(\alpha\, \mathrm{lq}^{(\alpha-1)} N\frac{\mathrm{lq}\,\mathrm{lq}\, N(1+\mathrm{lq}\, N)}{\mathrm{lq}^{\alpha} N}\right)$$
$$= O(\alpha\, \mathrm{lq}\,\mathrm{lq}\, N)$$

where the approximation (a) is due to the fact that $(1-e^{-y}) \approx y$ for small y; (b) follows from Binomial expansion; and (c) follows from the equality $\sum_{x > \alpha\, \mathrm{lq}\,\mathrm{lq}\, N} x^{m} q^{x} = O\left(\frac{m\, \mathrm{lq}\,\mathrm{lq}\, N}{\mathrm{lq}^{\alpha} N}\right)$, and the fact that the dominant term occurs when $m = 1$. Combining this result with (8) and the finding that $\hat{\Delta}$ is negligible, and then substituting these into (5) yields the expression for m_2^{RBC} stated in the the proposition. ■

Proposition 4 yields asymptotic formulations for the first and second moments of the maximum statistics of N Pascal distributed random variables of order K, and may, therefore, be of independent interest. Noting that we are primarily interested in understanding their effect on the mean waiting time, we next remark on the dominant terms. To that end, we study the dense network setting by fixing the file size K to a constant value and focusing on the asymptotic behavior as N increases. In this case, T behaves as $\mathrm{lq}\, N(1+o(1/\mathrm{lq}\, N)) \approx \mathrm{lq}\, N$ for large N. When this value is substituted in $m_{1,2}^{RBC}$ of Proposition 4, we can see that

$$m_1^{RBC} \approx \mathrm{lq}\, N, \qquad \text{and} \qquad m_2^{RBC} \approx \mathrm{lq}^2 N.$$

This is an interesting result when we note that $m_2^{RBC} \ge \left(m_1^{RBC}\right)^2$, due to Jensen's inequality. Thus, RBC asymptotically achieves the minimum possible second moment for the given first moment. Since we already know that m_1^{RBC} is the minimum achievable mean service time (cf. Proposition 1), this shows that RBC is also optimal in terms of minimizing the mean waiting time (cf. (1)).

2) Performance Analysis of RR: To compute the moments of the RR scheduler, we define X_k^i to be the number of transmissions of $\mathbf{P}_k$ before it is received by Receiver-i. Then,

$$Y^i \triangleq \max_{k \in \{1,\cdots,K\}} \{KX_k^i + k\}$$

gives the time slot when Receiver-i receives the whole file. Finally, $Z^{RR}(N,K) \triangleq \max_{i \in \{1,\cdots,N\}} Y^i$ gives the completion time of the RR scheduler. Similar to the RBC case, we use m_1^{RR} and m_2^{RR} to denote the first and second moments of $Z^{RR}(N,K)$, respectively. The next proposition provides tight bounds on m_1^{RR}.

Proposition 5. *Under symmetric channel conditions (i.e. $p_i = p \in (0,1)$ for all i), we have*

$$\frac{m_1^{RR}}{K} = \gamma + \sum_{t=1}^{\infty} \left[1 - \left(1 - q^t\right)^{KN}\right],$$

for some $\gamma \in (1/2, 1)$.

Moreover, the asymptotic performance of the moments of the RR scheduler with respect to N for fixed K satisfies

$$\frac{K}{2} + K \operatorname{lq}(KN) \leq m_1^{RR} \leq K + K \operatorname{lq}(KN),$$
$$m_2^{RR} \geq (K/2 + K \operatorname{lq}(KN))^2.$$

Proof: The upper bound of 1 for γ is due to the fact that $k \leq K$. The lower bound of $1/2$ follows from stochastic coupling arguments and heavily relies on the symmetry of the channel distributions. In particular, consider a sample path of the channel state process, $\omega \triangleq (\mathbf{C}[1], \mathbf{C}[2], \cdots)$. We use $i(\omega)$ to denote the receiver that was the last to complete the file, and $k(\omega)$ to denote the index number of the last packet that Receiver-$i(\omega)$ received. With our earlier notation, $Y(\omega)$ gives the completion time of the file at Receiver-$i(\omega)$ under the given sample path. Also, notice that we have $Y(\omega) = X_{k(\omega)}^{i(\omega)}(\omega)K + k$, for some integer $X_{k(\omega)}^{i(\omega)}$ that depends on ω.

Next, for each sample path ω that leads to $k(\omega) \in \{1, \cdots, \lfloor K/2 \rfloor\}$, we will construct another sample path $\tilde{\omega}$ that has the same probability of occurrence as ω, but leads to $Y(\tilde{\omega}) = X_{k(\omega)}^{i(\omega)}(\omega)K + (K - k(\omega))$. This implies that

$$\mathbb{E}[Y] \geq \frac{(K+1)}{2} + K\mathbb{E}[\max_{i,k} X_k^i]. \tag{13}$$

The construction of $\tilde{\omega} = (\tilde{\mathbf{C}}[1], \tilde{\mathbf{C}}[2], \cdots)$ follows the following rule:

$$\tilde{C}_j[rK + l] = \begin{cases} C_j[rK + (K - l)], & \text{if } r = X_{k(\omega)}^{i(\omega)}(\omega), \\ & j = i(\omega), \\ & l \in \{k(\omega), K - k(\omega)\}, \\ C_j[rK + l], & \text{otherwise.} \end{cases}$$

It can be seen that under symmetric conditions this sample path has the properties listed above.

Next, we would like to find the second term in (13). Due to i.i.d. assumptions, X_k^i are also i.i.d. with distribution $\mathbb{P}(X_k^i = m) = q^{m-1}p$, $m = 1, 2, \cdots$. Since this distribution is independent of i and k, we can compute

$$\mathbb{E}[\max_{i,k} X_k^i] = \sum_{t=1}^{\infty} \left[1 - \left(1 - q^t\right)^{KN}\right]. \tag{14}$$

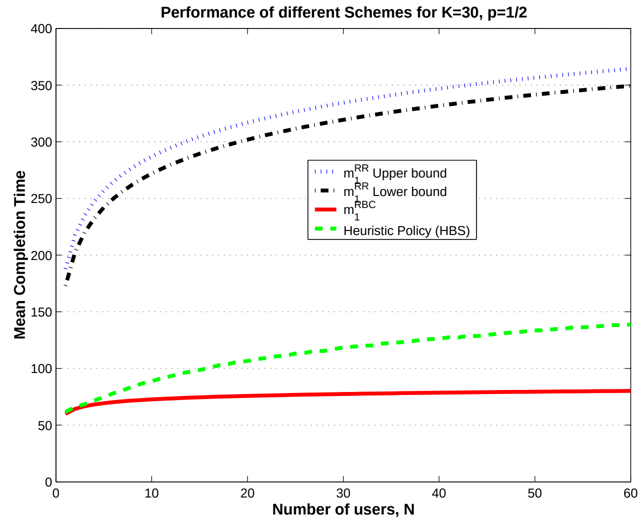

Fig. 2. Mean service time performance for $K = 30$ and $p = 1/2$.

The first part of the proof is complete once (14) is substituted into (13).

To prove the asymptotic expressions, we note that X_k^i is a Pascal distributed random variable of order 1. Therefore the derivation of m_1^{RBC} in Proposition 4 applies for computing $\mathbb{E}[\max_{i,k} X_k^i]$ with N replaced with KN, and K replaced with 1. To obtain m_2^{RR} we simply use Jensen's inequality: $m_2^{RR} \geq \left(m_1^{RR}\right)^2$. ■

C. Performance Comparison

In this section, we aim to demonstrate the coding gains on the mean waiting time for moderate and asymptotic values of N. To that end, we first provide numerical computations and simulations to compare the performance of various schemes we have discussed so far for moderate values of N and K. A comparison of the first moments of RBC, RR and HBS is illustrated in Figure 2 as a function of N, with $K = 30$, and each channel is ON or OFF equiprobably at every time slot.

The figure demonstrates the strength of the coding policy to the scheduling policy with and without CSI. We further observe that as N increases the advantage of using coding improves.

Figure 3 illustrates the waiting time performance of RBC versus RR for $K = 30$. It can be observed that the mean completion time gains are carried over to the mean waiting time performances. Notice that these huge gains are especially important to serve real-time traffic such as voice in unreliable networks.

Next, we provide the asymptotic gains of network coding compared to scheduling. We start by noting that for a fixed K, $m_1^{RBC} = \operatorname{lq} N(1 + o(1/\operatorname{lq} N))$ and $m_2^{RBC} = \operatorname{lq}^2 N(1 + o(1/\operatorname{lq}^2 N))$, whereas $m_1^{RR} = K \operatorname{lq} N(1 + o(1/\operatorname{lq} N))$ and

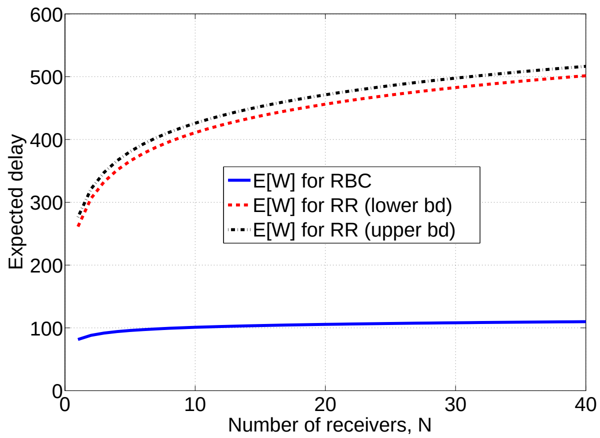

Fig. 3. Delay Performance of RBC versus RR for $K = 30$

$m_2^{RR} \geq K^2 \operatorname{lq}^2 N$. Substituting these in (1) yields

$$\overline{W}^{RBC}(\lambda) \sim \frac{\operatorname{lq}^2 N}{2(1-\lambda \operatorname{lq} N)}, \qquad \lambda < \frac{1}{\operatorname{lq} N}$$
$$\overline{W}^{RR}(\lambda) \sim \frac{K^2 \operatorname{lq}^2 N}{2(1-\lambda K \operatorname{lq} N)}, \qquad \lambda < \frac{1}{K \operatorname{lq} N}$$

We see that the maximum supportable arrival rate λ of RBC is K times that of RR. Moreover, when the system load is fixed to $\rho \in (0,1)$ fraction of the available capacity, i.e., $\lambda^{RBC} = \rho/\operatorname{lq} N$ and $\lambda^{RR} = \rho/(K \operatorname{lq} N)$, then we have[5]

$$\overline{W}^{RBC}(\lambda^{RBC}) = \frac{\overline{W}^{RR}(\lambda^{RR})}{K^2}.$$

In this section, we have seen that either with or without CSI, coding provides a considerable gain in the mean delay to download a given file to multiple receivers over a time-varying medium. Moreover, its operation is significantly easier than the scheduling policy. However, it requires an additional decoding operation at the receivers, which may or may not be critical depending on the file sizes and the computational capacity of the receivers.

IV. Serving Multiple Unicast Sessions

In this section, we consider the scenario where N receivers with symmetric channel conditions demand unique flows, i.e. $F = N$, and $N_f = 1$ for all $f \in \mathcal{F}$. In this case, it is not clear whether coding will have the dominating behavior as it did in the broadcast scenario. Again, the availability of CSI is important. In Section IV-A, we will study some of the properties of the optimal scheduling and coding strategies. Then, in Section IV-B, we will demonstrate the performance comparison through numerical computations.

A. Optimal Transmission Strategies

We will first study the scheduling case and then move on to the coding case.

[5]We note that the waiting time is measured per file, which consists of K packets.

1) Scheduling for Multiple Unicasts: We again consider the case of CSI and no CSI.

a) Scheduling without CSI:: Without CSI, the obvious optimal scheduling is again Round Robin, except that it must be performed across files and across packets in each file. In particular, in the first round the first packet of each file is transmitted one after another, and in the next round the second packets are transmitted consecutively. When the end of a file is reached, we move to the first packet and continue until all the packets of a file is received by its receiver. Only then we remove that file from the RR scheduler and continue with the remaining ones.

In this scenario, we define the completion time as the amount of time required for all the HOL files to be completed at the interested receivers. As before, we assume that only after all the HOL files are transmitted, are the transmission of the next batch of HOL files are starts. This model can be extended to give different weights to different flows, and hence achieve different fairness distributions. The mean completion time performance of the above RR scheduling rule is easy to compute using recursive arguments, which is omitted here since it does not add any significant insights to our analysis.

b) Scheduling with CSI:: Here, the constraint is to serve at most one receiver at every time slot. This problem is a special case of a problem studied by Tassiulas and Ephremides in [21] with no arrivals to the system. The following policy is introduced in [21].

LONGEST CONNECTED QUEUE (LCQ):
$t \leftarrow 0$;
$Q_i \leftarrow K_i$ for all $i \in \{1, \cdots, N\}$;
Do
  $t \leftarrow t+1$;
  $i^\star[t] \leftarrow \arg\max_{1\leq i\leq N}\{C_i[t]Q_i\}$;
  if($C_{i^\star}[t] \neq 0$)
    Transmit $\mathbf{P}_{i^\star, Q_{i^\star}}$;
    $Q_{i^\star} \leftarrow max(0, Q_{i^\star}-1)$;
While $\left(\sum_{i=1}^{N} Q_i > 0\right)$;
Return t; // Completion time

In the policy, Q_i is used both as a pointer to the index of the next packet to be transmitted to Receiver-i, and also as the number of packets yet to be transmitted to Receiver-i. Thus, LCQ is a myopic policy that favors the receiver with the maximum number of packets to be received among all connected receivers. We repeat the result of [21] for future reference.

Proposition 6 ([21]). *Under symmetric channel conditions (i.e. $p_i = p$ for all i), LCQ is minimizes the completion time over all scheduling policies. In other words,*

$$Z^{LCQ} \preceq_{st} Z^{\pi},$$

where T^{LCQ} denotes the completion time under the LCQ policy and π is any other feasible scheduling policy[6].

This result is very strong and implies that $\mathbb{E}[Z^{LCQ}] \leq \mathbb{E}[Z^{\pi}]$ for any feasible scheduling policy π.

[6]$\preceq_{st}$ is a stochastic ordering as described in [21].

2) Coding for Multiple Unicasts: A deep understanding of achievable rates for multiple unicast sessions in a network is still an open problem. In general, it is not clear whether network coding should be performed, and if it should what the strategy must be. We will tackle this problem for the downlink model at hand.

We define the set of *coding classes* that partitions $\mathcal{F}$ (or equivalently $\mathcal{N}$) into J subsets. We use $\mathcal{C}_j$ to denote the files (or equivalently receivers) in Class-j. We set the restriction that only those files within the same class will be linearly coded with random coefficients as in RBC, while files of different classes will not be mixed. Notice that for each class, say $\mathcal{C}_j$, this strategy effectively results in a single file of length $K^j \triangleq \sum_{f \in \mathcal{C}_j} K_f$ that is demanded by $b_j \triangleq |\mathcal{C}_j|$ distinct receivers. Hence, the multiple unicasts problem is converted into a special case of multiple multicasts with each multicast having a disjoint set of receivers. Notice that the description of the strategy is yet incomplete, because we must describe how to "schedule" the transmissions of different classes. We will investigate this question with and without CSI.

a) Coding without CSI: In this case, as in Section III-A.1, we assume that each receiver informs the transmitter when it can decode its own file, which in turn implies that it can decode all the files within its class. The optimal policy is again going to be of the form of Round Robin over the coding classes. We will consider the case of $b_j = b$ and $K^j = \tilde{K}$ equal for all j. If J denotes the total number of coding classes, then only a combination from $\mathcal{C}_j$ will be transmitted in slot $(mJ+j)$ for $m = 0, 1, \cdots$ until all the receivers get their files.

Notice that the analysis of the RR scheduler of Section III-A.1 does not directly apply to this case, because here once all the receivers of a class, say $\mathcal{C}_j$, decode their file, then that class can be extracted from the round robin cycle. Nevertheless, similar analysis based on recursive formulations can be used for this setting. This analysis is omitted here due to space constraints. We remark that without CSI the gain in grouping subsets of users as described above is only due to the decreasing size of the cycles as groups complete their receptions. If the period of each cycle were kept constant at its starting value of J throughout the operation, then grouping would have no effect on the average delay performance, because in such a scenario we would be comparing the expected number of slots before K ON channels are observed to $1/b$ times the expected number of slots before bK ON channels are observed.

b) Coding with CSI: In the presence of CSI, we must determine the optimal partitioning of the files $\{\mathcal{C}_j\}$, and also find the optimal scheduling policy across these classes. The following proposition finds the optimal policy using stochastic coupling arguments.

Proposition 7. *Under the symmetric channel conditions (i.e. $p_i = p$ for all $i \in \mathcal{N}$), the mean delay minimizing partitioning is obtained when $b_j = 1$ for all j, and the optimal policy is to implement LCQ.*

Proof: Consider any given partitioning of the files, say $\mathcal{P} = \{\mathcal{C}_j\}_{j=1}^J$, and let $\pi_{\mathcal{P}}$ denote the optimal policy for this partitioning, which is not known in general. Also, let $T^{\pi_{\mathcal{P}}}$ be the random variable that denotes the completion time of all the files under the policy $\pi_{\mathcal{P}}$. In other words, $T^{\pi_{\mathcal{P}}}$ is the first slot when each receiver in Class-j received K^j linear combinations of the packets from within their class, for all j. We use $\omega = (\mathbf{C}[1], \mathbf{C}[2], \cdots)$ to denote a sample path of the channel state process. Notice that the policy and ω determines $T^{\pi_{\mathcal{P}}}(\omega)$.

Next, we will define a new policy $\tilde{\pi}$ and show that it satisfies $T^{\tilde{\pi}}(\omega) \leq T^{\pi_{\mathcal{P}}}(\omega)$ for all feasible ω. For a given ω, if $\pi_{\mathcal{P}}$ serves Class-j in slot t, then $\tilde{\pi}$ will send only the head-of-line packet of one of the connected receivers in the same class which received the minimum service so far. In other words, amongst the connected receivers in Class-j, only the receiver that has the maximum number of remaining packets is served. Notice that this policy does not do any coding, and hence requires Receiver-f in Class-j to successfully receive K_f packets of its file instead of K^j packets as in $\pi_{\mathcal{P}}$.

To see that $T^{\tilde{\pi}}(\omega) \leq T^{\pi_{\mathcal{P}}}(\omega)$, observe that whenever Class-j is served under $\pi_{\mathcal{P}}$, at most one packet (or one degree of freedom) can be received by each receiver in that class. Thus, before all of its receivers can decode their own packet, Class-j must be served at least K^j times. But, with $\tilde{\pi}$ we can send a single degree of freedom to one of the connected receivers in Class-j whenever that class is served under $\pi_{\mathcal{P}}$. Since for each $f \in \mathcal{C}_j$, only K_f degrees of freedom are required for Receiver-f with $\tilde{\pi}$, all the receivers complete their reception when Class-j is served $K^j = \sum_{f \in \mathcal{C}_j} K_f$ times. These arguments prove that for any feasible sample paths the completion of the new policy is not larger than that of $\pi_{\mathcal{P}}$ for any partition $\mathcal{P}$.

To complete the proof, we need to show that $T^{LCQ} \preceq T^{\tilde{\pi}}$. To that end, we note that $\tilde{\pi}$ is actually a scheduling policy, where at each slot a single packet is transmitted over the channel. Thus, an application of Proposition 6 completes the proof. ■

B. Performance Comparison

In this section, we compare the typical performance of various policies for reasonable parameters. We take $b_j = b$ for all j and $K_f = K$ for all $f \in \mathcal{F}$. Moreover, we let $K = 30$ and $N = F = 12$ and study the mean completion time behavior of the scheduling and coding strategies with and without CSI. Regarding the channel connectivity statistics, we assume that $p_i = 1/2$ for all the channels. Figure 4 depicts the simulation results of the policies discussed above for varying number of classes. In the figure, we observe that the performance of the LCQ scheduler serves as a lower bound as we have proved in Proposition 7. Since the optimal coding policy is not specified for an arbitrary b, in the simulation we use the following heuristic policy: at each time slot among the classes with the maximum number of connected receivers, the policy serves the class with the maximum degrees of freedom yet to be transmitted. This policy, when $b = 1$ is the same as the LCQ policy. For this policy, we observe that the mean delay value achieved decreases to half its value when b is decreased from 12 to 1. We also observe that in agreement with our arguments, the performance of the coding without CSI improves as b decreases, but this decrease is rather insignificant.

Without CSI, the performance of scheduling is significantly worse than the coding solution. In this particular case, we

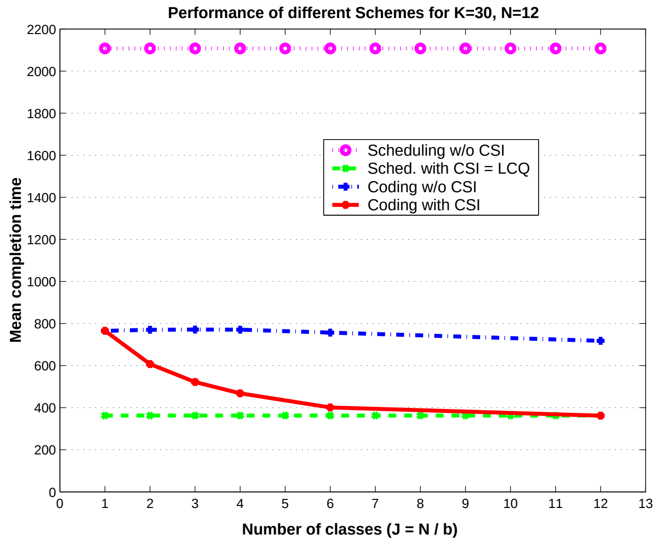

Fig. 4. Mean completion time of various strategies for $p = 1/2$.

observe almost a threefold delay with scheduling as opposed to coding. Given that the single-hop multiple unicasts scenario does not improve the capacity of the channel, the presence of such a considerable delay gain is particularly striking.

The fact that both with and without CSI the performance of the coding strategy improves as b goes to one implies that for unicast transmissions, it is best to code within files, but not across them.

V. Extension to General Topologies

So far we have considered single-hop wireless networks in which packets are transmitted from the transmitter to each receiver over a single hop without any intermediate relaying mechanism. We next present a simple model to study delay gains from coding in multi-hop wireless networks. We achieve this by rearranging the general topology into a layered topology, and then analyzing the layered topology as a chain of single-hop networks. The following example demonstrates our layering approach.

Example 2 (Decomposition of a Network into Layers): Consider the multicast setting shown in Figure 5 consisting of two sink nodes, a single source node and some intermediate nodes. We decompose the network in layers such that a node belongs to Layer-i if the shortest path from the source to it is i hops. We can identify the layer in which each node is to be placed by simply flooding the network or by using sophisticated shortest path algorithms. The files generated at the source are transmitted from one layer to the next subject to interference constraints that will be discussed next. ◇

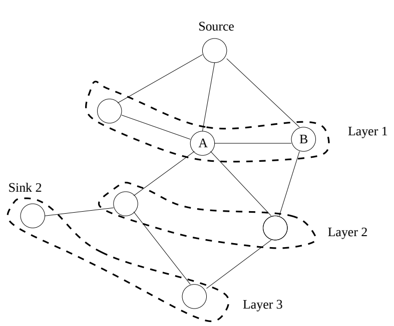

Fig. 5. A multicast setting in a general network topology

We assume that nodes belonging to the same layer are scheduled to transmit at the same time, while nodes in different layers transmit in orthogonal channels (e.g., disjoint frequency bands or time slots). Therefore, only the transmission of nodes belonging to the same layer interfere. The interference model is assumed to be a collision channel for each receiver, i.e., a receiver successfully receives a packet in a time slot if only if it receives exactly one packet in that time slot. We assume that there is no communication among nodes within the same layer, i.e., we drop all links among nodes within the same layer. Therefore, both node A and node B are placed in Layer 1 in Figure 5, and the link between A and B is dropped.

The next step is to analyze the layered network as a series of single-hop networks. The source transmits the file to the first layer, the first layer transmits the file to the second layer, and so on, until the file reaches all the sink nodes. Sinks can be in different layers, in which case the layer-to-layer transmission will end when the file is received by the sink node in the last layer. Note that packet transmission between two adjacent layers is identical to the single-hop case described previously, with two important differences; first, both the transmitting layer and receiving layer may have more than one node (i.e., there can be multiple transmitters and multiple receivers), and second, the presence of multiple transmitting nodes may lead to collisions at the receivers. Therefore, before using results from the single-hop case, we must extend the single-hop case to to model multiples transmitters and multiple receivers. This extension is described next.

A. Multiple-transmitter Multiple-receiver Systems

Consider a single layer with N_s transmitters and N_r receivers. Transmissions take place in regularly arranged time slots with one packet per time slot. Assume for simplicity that each receiver is linked to a randomly chosen subset of the transmitters, and that the cardinality of the subset, denoted by L, is the same for each receiver, i.e., all receivers are connected to an equal number of transmitters. This is the symmetric case. In the asymmetric case, each receiver is allowed to be connected to a different number of transmitters. The channel conditions on each link are identical to the channel conditions, i.e., each channel is ON with probability p in each time slot or OFF otherwise. Figure 6 illustrates the system topology for $N_s = 3$ and $N_r = 4$. Here, each receiver is connected to two transmitters.

Initially, all transmitters have the same file consisting of K packets. Our goal is to minimize the time taken for the file to be transmitted to all the receivers, and to compare the mean file transfer completion times for network coding and scheduling in the presence of multiple transmitting nodes.

Since transmission is successful only if a receiver receives one packet in a time slot, it does not make sense for each transmitter to transmit in every time slot. In the absence of

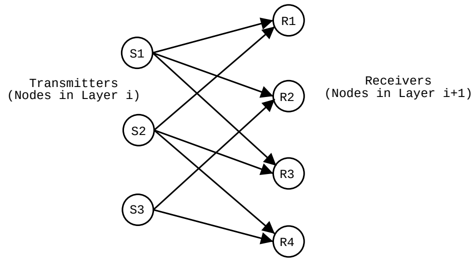

Fig. 6. A multiple-transmitter multiple-receiver system with three transmitters and four receivers

communication among transmitters, a better strategy is for transmitter S_i to attempt transmission with probability c_i in every time slot. For simplicity, we restrict our attention to the symmetric case in which $c_i = c$ for all transmitters. Since the channel between S_i and, say, receiver R_i is ON with probability p, the probability that R_i successfully receives a packet from S_i is pc. Recalling that L transmitters are connected each receiver, the number of packets a receiver receives in one time-slot, X, is given by a binomial distribution with parameters (L, pc). Hence, the probability that a given receiver successfully receives a packet in a time slot is $\mathbb{P}(X = 1) = Lpc(1-pc)^{L-1}$. This expression is identical to the probability of a successful capture in the Aloha system. It is well-known that the optimal reception probability pC must be $1/L$, yielding a success probability of approximately $1/e$. Therefore, the number of packets a receiver receives in one time slot is Bernoulli distributed with a success probability of $1/e$. Thus, the results of Section III directly applies to find the mean completion times for network coding and scheduling by replacing $p = 1/e$ and $N = N_r$.

VI. Conclusions

In this work, we introduced a key setting where delay performance of network coding can be investigated and compared to the traditional method of scheduling. Under various scenarios, we identified the optimal policies and derived analytical expressions for the delay expressions. We provided explicit characterization of the delay performance achieved by coding and scheduling both for moderate and asymptotic values of system parameters. Our findings reveal the significant delay gains of coding in unreliable networks. Moreover, we pointed to ways of extending our results to cover more general network settings.

These fundamental findings has interesting implications on the performance of the applications at the higher layer of the network hierarchy. There are numerous problems of interest based on the findings of this work. For example, how does the delay gains revealed in this work reflect to a scenario where users have delay constraints? Also, are there more efficient ways of implementation in the general network scenario? What is the tradeoff between delay performance and overhead? We aim to address some of these problems in our future research.

Acknowledgments

Authors would like to thank Peter Grabner for his valuable comments and contributions in the proof of Proposition 4.

References

[1] R. Ahlswede, Ning Cai, S. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46:1204–1216, July 2000.

[2] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, Englewood Cliffs, NJ, 1987.

[3] S. Bhadra and S. Shakkottai. Looking at large networks: Coding vs. queueing. In *Proceedings of IEEE Infocom*, Barcelona, 2006.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. M.I.T. Press, McGraw-Hill Book Company, London, England, 2001.

[5] A. Eryilmaz, A. Ozdaglar, and M. Médard. On delay performance gains from network coding. In *Proceedings of Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, March 2006.

[6] M. Ghaderi, D. Towsley, and J. Kurose. Reliability benefit of network coding. Technical Report, Computer Science Department, University of Massachusetts Amherst, February 2007.

[7] P. J. Grabner and H. Prodinger. Maximum statistics of n random variables distributed by the negative binomial distribution. *Combinatorics, Probability and Computing*, 6:179–183, 1997.

[8] T. Ho. Networking from a network coding perspective. PhD thesis, MIT, 2004.

[9] D. Katabi, S. Katti, W. Hu, H. Rahul, and M. Médard. The importance of being opportunistic: On practical network coding for wireless environments. In *Proceedings of Annual Allerton Conference on Communication, Control and Computing*, October 2005.

[10] D. Katabi, S. Katti, W. Hu, H. Rahul, and M. Médard. On practical network coding for wireless environments. In *Proceedings of IEEE International Zurich Seminar on Communications*, 2006.

[11] R. Koetter and M. Médard. Beyond routing: An algebraic approach to network coding. *IEEE Transactions on Information Theory*, 11:782–795, October 2003.

[12] S.-Y. R. Li, R. W. Yeung, and Ning Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49:371–381, February 2003.

[13] D. S. Lun, M. Médard, R. Koetter, and M. Effros. Further results on coding for reliable communication over packet networks. In *Proceedings of IEEE International Symposium for Reliable Communication over Packet Networks*, pages 1848–1852, 2005. (An extended version has been submitted to the *IEEE Transactions on Information Theory*).

[14] D. S. Lun, P. Pakzad, C. Fragouli, M. Médard, and R. Koetter. An analysis of finite-memory random linear coding on packet streams. In *Proceeding of NetCod*, 2006.

[15] D. S. Lun, N. Ratnakar, R. Koetter, M. Médard, E. Ahmed, and H. Lee. Achieving minimum cost multicast: A decentralized approach based on network coding. In *Proceedings of IEEE Infocom*, pages 1607–1617, 2005.

[16] D. Nguhyen, T. Nguyen, and B. Bose. Wireless broadcasting using network coding. In *Proceeding of NetCod*, 2007.

[17] R. B. Paris and D. Kaminski. *Asymptotics and Mellin-Barnes integrals*, volume 85 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2001.

[18] S. Shakkottai P. Gupta S. Bhadra. Min-cost selfish multicast with network coding. In *Proceeding of NetCod*, 2006.

[19] B. Shrader and A. Ephremides. On the queueing delay of a multicast erasure channel. In *IEEE Information Theory Workshop (ITW)*, October 2006.

[20] J. Sundararajan, D. Shah, and M. Médard. Queueing advantages of network coding, 2007. Technical Report.

[21] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information Theory*, 39:466–478, March 1993.

[22] B. S. Tsybakov. File transmission over wireless fast fading downlink. *IEEE Transactions on Information Theory*, 48:2323–2337, August 2002.

[23] Yunnan Wu. Network coding for multicasting. PhD thesis, Princeton University, November 2005.

[24] Y. Xi and E. M. Yeh. Distributed algorithms for minimum cost multicast with network coding in wireless networks. In *Second Workshop on Network Coding, Theory, and Applications*, Boston, MA, April 2006.