On the Scaling Law of Network Coding Gains in Wireless Networks

Ebad Ahmed, Atilla Eryilmaz, Muriel Médard, and Asuman E. Ozdaglar
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA, 02139
Emails: {ebad,eryilmaz,asuman,medard}@mit.edu

***Abstract*— We study the scaling law governing the delay gains of network coding as compared to traditional transmission strategies in unreliable wireless networks. We distinguish between two types of traffic, namely *elastic* and *inelastic*, where the elasticity of a flow is based on the delay constraints associated with it. This novel formulation is useful in that it allows for the modeling of real-time traffic more accurately. Considering the limited availability of feedback in such systems, we focus on strategies with minimal acknowledgement requirements. Under both traffic types, we provide an extensive analysis of the gains of network coding as compared to traditional transmission strategies in a single-hop setting, and show that the gains are significant in general and can be considerably large in some cases. We further provide a method for realizing these gains in multi-hop networks with general topologies using the analysis of the single hop scenario.**

I. Introduction

The problem of developing practical transmission strategies in interference-limited networks has been addressed in numerous recent works (e.g. [9], [12], [4], [10]) with focus on throughput performance. These strategies aim to find ways to schedule transmissions to avoid collisions under the assumption of no channel fading. Yet in reality channel fluctuations often occur because of mobile nodes and other obstacles in the environment. For existing strategies to be usable in such conditions, the channel state information (CSI) must be available to the relevant nodes at the outset of transmission. This, in general, is a costly and difficult condition to guarantee. Furthermore, in many situations, the multicast set is large and it is impractical to receive acknowledgements from all the receivers. Thus, we focus on the case of no CSI at the transmitters. The case with acknowledgements has been considered in [5], where an ARQ-based feedback scheme is studied.

An important factor in many real-world systems is delay: networks with large delays become infeasible to operate. Delay performance is even more critical than throughput performance in the presence of real-time traffic such as voice and video. With this view, we distinguish between *elastic* and *inelastic* traffic by introducing a novel utility-based formulation. We assume that every block of K packets (also called a *user*) represents a file with a certain delay demand. A user derives utility from service only if it is completed within its associated delay threshold. For elastic traffic there is no delay constraint, while for inelastic traffic the users prefer not to get any service if the expected delay is above a certain threshold. Note that this delay-based utility formulation is different from those in the networking literature (see [11] and the references therein) where utility is traditionally defined as a function of the flow rate. Our formulation has the advantage of allowing a more accurate modeling of delay-constrained real-time traffic.

This work builds on [3], which introduced a network-coding-based transmission strategy as an alternative to the traditional scheduling strategy and showed its delay advantages in a single-hop setting. We model a dynamic system with a stream of incoming files to be transferred over unreliable channels to multicast receivers. We provide asymptotic expressions on the delay performance of the two strategies in the limit of a sufficiently large number of receivers in the multicast set, and show that network coding can asymptotically perform arbitrarily better than scheduling.

We further show that the single-hop setting can serve as the fundamental building block for more general network topologies. Its extension to tree topologies has already been considered in [5]. In this work, we propose a method for utilizing the single-hop arguments in a more general topology by transforming the network into a sequence of layers whereby each layer can operate independently from the rest. We show how this layered model can in turn be analyzed using the results obtained for the single-hop case, and how the corresponding gains extend to this scenario.

The rest of the paper is organized as follows. In Section II, we describe our system model. In Section III, we characterize the delay gains from network coding as compared to scheduling under both elastic and inelastic traffic. In Section IV, we delineate the extension of the single-hop network to general multicast topologies and investigate the delay gains from network coding. We conclude in Section V.

II. System Model

In this section, we describe the single-hop setting that will serve as the fundamental building block for more general networks. The connection to general topologies will be made explicit in Section IV.

This work is supported by the Control-Based Mobile Ad-Hoc Networking (CBMANET) Program under DARPA subcontract no. 060786.

Consider a node broadcasting a sequence of incoming files to N neighboring receivers over time-varying channels. The files that have entered the queue are served in a First-In-First-Out (FIFO) fashion. Thus, the transmission of the next file starts after the current file has been received by every receiver. Each *file* is assumed to be composed of K *packets*, where Packet-k of a given file is referred to as $\mathbf{P}_k$, which is a vector of length m over a finite field $\mathbb{F}_q$, for some $q \in \mathcal{Z}_+$. Transmissions take place in regularly arranged time slots with each slot long enough to accommodate a single packet transmission. The channel between the transmitter and each receiver has a time-varying nature to capture the influence of fading, interference and mobility of the receivers. Specifically, we assume that the channel condition in slot t between the transmitter and the n^{th} receiver is captured by a Bernoulli distributed random variable $C_n[t]$ with mean c_n that is independent across users and time slots. When $C_n[t] = 1$, the channel is assumed to be ON and the packet sent by the transmitter is successfully received by the n^{th} receiver. If, on the other hand, $C_n[t] = 0$, the transmitted packet in slot t does not reach receiver n. We will refer to c_n as the mean channel rate for channel n. In general, Channel Side Information (CSI, i.e., the state of each channel at the outset of transmission) may or may not be available to the transmitter. However, unless N is very small, the assumption of the availability of CSI is impractical because of the requirements of frequent feedback and training signals. Therefore, we focus on the realistic scenario where no CSI is available to the transmitter, and feedback is sent only when a receiver gets the whole file[1]. Such a system is not only simpler to implement but also dissipates less energy and bandwidth resources for overhead signals.

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 now define the two transmission strategies that we will analyze in this paper.

Definition 1 (Scheduling): Let $\mathbf{P}[t]$ denote the packet chosen for transmission in slot t. If the transmitter is not allowed to code, then at any given slot it must transmit a single packet from the current file. Thus, we have $\mathbf{P}[t] \in \{\mathbf{P}_k\}_{\{k=1,\cdots,K\}}$. We will refer to this mode of transmission as the *Scheduling Mode* (or simply *Scheduling*). ◇

Definition 2 ((Network) Coding): If coding is allowed, then in a slot, say t, any linear combination of the K packets in the file can be transmitted. Specifically, we have

$$\mathbf{P}[t] = \sum_{k=1}^{K} a_k[t]\mathbf{P}_k,$$

where $a_k[t] \in \mathbb{F}_q$ for each $k \in \{1, \cdots, K\}$. The transmitter chooses the coefficients $\{a_k[t]\}$ at every time slot t. We will henceforth refer to this mode of transmission as the *(Network) Coding Mode* (or simply *Coding*). ◇

[1]The effect of CSI availability is studied in detail in [3], where it has been observed that coding is optimal even when no CSI is available.

This setting was first introduced in [3] along with an analysis of file completion time performance in various situations. It is also discussed in [3] that the overhead associated with the transmission of the coefficient information is negligible for large field sizes. In this paper, we extend the model in [3] by providing an asymptotic performance analysis of coding and scheduling, and by modeling more general network topologies. Moreover, we investigate the effects of delay-constrained traffic on the file download completion time. We consider two types of traffic: elastic and inelastic. These are defined next.

Definition 3 (Elastic Traffic): Elastic traffic refers to a stream of incoming files (also called *users*) with no delay constraints. For this type of traffic, each user derives a fixed amount of utility from the service regardless of the time it takes for the service to be completed. In other words, all users are willing to join the system regardless of how large a queuing delay they will encounter upon accepting service. ◇

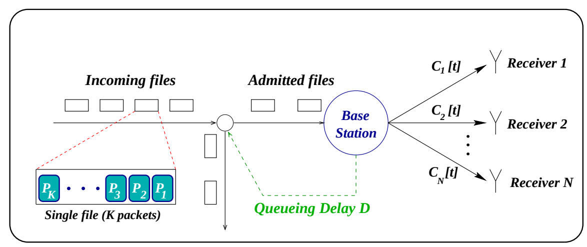

Fig. 1. System model.

Definition 4 (Inelastic Traffic): Inelastic traffic refers to the case in which users (files) have stringent delay constraints and enter the system only if their delay constraints are guaranteed to be met. In particular, when the delay constraint is, say, d_{avg}^{max}, users enter for service only if the transmitter can guarantee a mean file completion time of d_{avg}^{max} or less. Otherwise the users leave the system (see Figure 1). ◇

Note that these definitions of elastic and inelastic traffic are different from those traditionally used in the network optimization framework in that our formulation associates utilities with delay rather than throughput. This is an important modification, especially because it allows us to model delay-constrained real-time traffic such as voice and video broadcast. In subsequent sections, we will analyze the performance of the system under both types of traffic using the economic framework introduced in [2].

III. Asymptotic Performance

Our goals in this section are first, to investigate the asymptotic mean delay performance (as N and K scale) of network coding versus scheduling under elastic traffic, and second, to study the performance in a utility-based framework with inelastic traffic. We first state some preliminary results which we will later use in our analysis.

A. Preliminaries

It has been shown in [3] that the optimal strategy under the scheduling mode is the *Round-Robin (RR) strategy*, in which packets of the current file are transmitted in order periodically until all the receivers receive all the packets. The optimal strategy under the coding mode is the *Random Network Coding (RNC) strategy*, where in every time slot $a_k[t]$ is chosen uniformly at random from $\mathbb{F}_q \backslash \{0\}$. Therefore, our subsequent analysis is based on the RR and RNC strategies.

Let T_{RNC} and T_{RR} denote the file download completion times for RNC and RR strategies, respectively. The following results are from [3].

Proposition 1: Under symmetric channel conditions, i.e., $c_i = c$ for all $i \in \{1, \cdots, N\}$,

(A) $T_{RNC} = \max_{1 \le i \le N} Y_i$, where the Y_i's are *i.i.d* random variables following a Pascal distribution of order K and parameter c, and K is the file size.

(B) $T_{RR} = \max_{1 \le i \le N} \max_{1 \le k \le K} \{K W_i^k + k\}$, where the W_i^k's are *i.i.d* geometric random variables with parameter c.

(C) The mean of T_{RNC}, $\mathbb{E}[T_{RNC}]$, is given by

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

where $\bar{c} = (1 - c)$, $\begin{pmatrix} n \\ m \end{pmatrix}$ gives the number of combinations of size m of n elements.

(D) For some $\gamma \in (1/2, 1)$, $\mathbb{E}[T_{RR}]$ satisfies

$$\frac{\mathbb{E}[T_{RR}]}{K} = \gamma + \sum_{t=1}^{\infty} \left[1 - (1 - (1-c)^t)^{KN} \right].$$

Although the mean file completion times as given by Proposition 1-(C),(D) are exact, they are not explicit, closed-form functions of N and K, which makes it impossible to acquire a qualitative understanding of the dependence of $\mathbb{E}[T_{RNC}]$ and $\mathbb{E}[T_{RR}]$ on N and K. Proposition 2 provides more tractable expressions for $\mathbb{E}[T_{RNC}]$ and $\mathbb{E}[T_{RR}]$ which provide a better understanding of how the mean completion times depend on N and K.

Proposition 2: Let us use $\mathrm{lc}(\cdot)$ as a shorthand for $\log_{\left(\frac{1}{1-c}\right)}(\cdot)$. Under symmetric channel conditions, we have

$$\begin{aligned} \mathbb{E}[T_{RNC}] &= \mathrm{lc}\, N + (K-1)\, \mathrm{lc}\, \mathrm{lc}\, N + (K-1) \\ &\quad - \mathrm{lc}\,((K-1)!) + o(\log N), \end{aligned} \tag{1}$$

and

$$\frac{K}{2} + K\, \mathrm{lc}\, KN \le \mathbb{E}[T_{RR}] \le K + K\, \mathrm{lc}\,(KN). \tag{2}$$

Proof: Equation (1) follows from [6] since T_{RNC} is the maximum of N Pascal variables of order K. Equation (2) follows from (1) after noting that a geometric random variable is a Pascal random variable of order 1. ■

B. Asymptotic Performance Analysis

We now provide an analysis of the asymptotic performance of the RNC and RR strategies under elastic and inelastic traffic.

1) Elastic Traffic: We use $G(N,K) \triangleq \frac{\mathbb{E}[T_{RNC}]}{\mathbb{E}[T_{RR}]}$ as a measure of the relative delay gains from network coding as compared to scheduling. Since there are no delay constraints under elastic traffic, the behavior of $G(N,K)$ in the limit as $N \to \infty$ will give the asymptotic delay gain from network coding as the number of receivers increases. The following result shows that the relative gain of network coding grows linearly with K.

Proposition 3: For a fixed file size of K, the relative gain $G(N,K)$ of network coding compared to scheduling has the following asymptotic behavior:

$$\lim_{N \to \infty} G(N,K) = \frac{1}{K}. \tag{3}$$

Proof: In order to get an upper bound on $G(N,K)$ (i.e., delay gains from network coding in the worst-case scenario), we use the lower bound on $\mathbb{E}[T_{RR}]$ from (2), which yields

$$G(N,K) \le \frac{\mathrm{lc}\, N + (K-1)\, \mathrm{lc}\, \mathrm{lc}\, N + o(log(N))}{\frac{K}{2} + K\, \mathrm{lc}\,(KN)}.$$

Similarly, a lower bound on $G(N,K)$ can be achieved using the upper bound on $\mathbb{E}[T_{RR}]$ in (2). If we fix the file size K and consider the limiting case $N \to \infty$, then for both the upper and lower bounds, the dominant term in the numerator is $\mathrm{lc}\, N$, and the dominant term in the denominator is $K\, \mathrm{lc}\, N$. This completes the proof. ■

The asymptotic ratio of the the file completion time of network coding to that of scheduling is the reciprocal of the file size K, which signifies that as the number of receivers increases to a sufficiently large value, file downloads take K times longer if scheduling is used instead of network coding. In general, larger relative asymptotic gains from network coding can be realized for larger file sizes. Thus, although the gain is fixed for fixed values of K, it is essentially unbounded depending on the value of K.

It should be noted that the larger value of K increases the computational complexity required for the decoding operation. Thus, in the choice of K, the tradeoff between delay and complexity needs to be taken into account. Also, in many cases the higher order moments of the delay (measuring delay jitter) are of critical importance to the quality of real-time communication. The corresponding analysis requires a deep study of the asymptotic behavior of extremal distributions, which is part of our future work. Our preliminary investigations suggest that network coding exhibits lower variance than scheduling for file transmission time, hence promising smaller delay jitter at the receivers.

2) Inelastic Traffic: The gain expression in the previous section was obtained under the the assumption of elastic traffic, i.e., users were willing to join the system regardless of the queueing delay they had to face. This is often not the case in practice. In the more realistic scenario where the demand for file download is inelastic [cf. Definition 4], the above scaling law, $\lim_{N \to \infty} G(N,K) = \frac{1}{K}$ ceases to hold. In this case, the more relevant questions to consider are:

• Given a fixed queueing delay constraint of d_{avg}^{max} time slots, a fixed file size K, and a fixed user admission rate λ, how

many more receivers (i.e., N) can the system support with network coding than with scheduling?
• How does N scale with d_{avg}^{max}, i.e., how does relaxing the delay constraint affect the number of users the system can support?
• Given a fixed N and a fixed K, how does λ scale with d_{avg}^{max}, i.e., how does relaxing the delay constraint affect the user admission rate or the throughput of the system?
• Given a fixed N and a fixed d_{avg}^{max}, how does λ scale with K, i.e., how does changing the file size affect the throughput of the system?

In order to answer these questions, we formulate the problem within the following utility-based economic framework. Users (or files) arrive with a rate of $\gamma > 0$ (the offered load to the system) to be broadcast to all the N receivers. Each user has a delay constraint Θ is uniformly distributed between 0 and d_{avg}^{max}. This constraint captures the delay requirements associated with the users. A user draws a utility of 1 from the service if transmission is completed in less than Θ time slots, and derives a utility of 0 otherwise[2]. Each user has the option of either accepting or refusing service based on whether or not it expects its delay constraint to be met.

We now analyze the performance of network coding and scheduling under various scaling laws in this setting. The system can be modeled as a $G/G/1$ queue. In order to derive expressions that represent the general case, we focus on the case in which the arrival process is Poisson and model the system as an $M/G/1$ queue. For other arrival processes (such as the deterministic arrival process), various bounds such as Kingman's bound [8] can be used to characterize the system delay.

In the $M/G/1$ case, each user will experience a delay $D(\gamma, N, K)$ depending on the transmission strategy used by the transmitter (RNC or RR) and the number of users waiting in the queue (dictated by the arrival rate γ). The expression for the expected delay is given by the *Pollaczek-Khinchin* formula:

$$\mathbb{E}[Delay] = \frac{\lambda \mathbb{E}[Z^2]}{2(1-\lambda \mathbb{E}[Z])}, \qquad (4)$$

where Z is the service time of a single file broadcast. The distribution of Z will depend on the transmitter's transmission strategy. Henceforth, we will use m_1 and m_2 to denote $\mathbb{E}[Z]$ and $\mathbb{E}[Z^2]$ respectively. We characterize the user admission rate λ in the following proposition.

Proposition 4: Let the arrival process be Poisson with rate γ and the delay constraint for a typical user be uniformly distributed between 0 and d_{avg}^{max}. Define $h = h(\gamma, d_{avg}^{max}, m_1, m_2) = 2\gamma d_{avg}^{max} m_1 + \gamma m_2 + 2d_{avg}^{max}$. Then the user admission rate λ takes the following form:

$$\lambda = \frac{h - \sqrt{h^2 - 16(d_{avg}^{max})^2 \gamma m_1}}{4 d_{avg}^{max} m_1}$$

Proof: Let the queuing delay be $\mu(\lambda)$ at any given instant. Each user will decide to enter if and only if its net utility from the file transfer is non-negative, which will be the case only when the user's delay constraint $\Theta \geq \mu$. This implies that the *effective input rate* or *accepted load* λ is given by

$$\lambda \;=\; \gamma \mathbb{P}(\Theta \geq \mu) = \gamma \int_{\mu}^{d_{avg}^{max}} \frac{1}{d_{avg}^{max}} \, d\theta.$$

We therefore have

$$\lambda = \begin{cases} \gamma \frac{d_{avg}^{max} - \mu}{d_{avg}^{max}} & \text{if } \mu < d_{avg}^{max} \\ 0 & \text{if } \mu \geq d_{avg}^{max} \end{cases} \qquad (5)$$

For the case where $\mu < d_{avg}^{max}$, we have

$$\lambda \;=\; \gamma \left(1 - \frac{\mu}{d_{avg}^{max}}\right) = \gamma \left(1 - \frac{\lambda m_2}{2 d_{avg}^{max} \left(1 - \lambda m_1\right)}\right),$$

using the *Pollaczek-Khinchin* formula (4). Algebraic manipulations yield the following quadratic equation in λ,

$$2d_{avg}^{max} m_1 \lambda^2 - (2\gamma d_{avg}^{max} m_1 + \gamma m_2 + 2d_{avg}^{max})\lambda + 2\gamma d_{avg}^{max} = 0,$$

with roots

$$\lambda_{1,2} \;=\; \frac{h \pm \sqrt{h^2 - 16(d_{avg}^{max})^2 \gamma m_1}}{4 d_{avg}^{max} m_1},$$

where $h = h(\gamma, d_{avg}^{max}, m_1, m_2)$ as defined in Proposition 4. In order to satisfy the constraint that the delay μ is finite and positive, we must have $\lambda < \frac{1}{m_1}$ (so that the term $(1-\lambda m_1)$ in the *Pollaczek-Khinchin* formula (4) is positive). The proof is complete when the infeasible root in the previous expression is eliminated. ■

The characterization of the user admission rate λ given by Proposition 4 involves m_1 and m_2, which are functions of N and K. Therefore, in order to observe the dependence of N on d_{avg}^{max} using (4), K and λ must be held constant and d_{avg}^{max} must be varied. We now derive explicit expressions to compute m_1 and m_2 for both the RNC and RR strategies. Let us introduce the superscripts RNC for network coding and RR for round-robin scheduling. Henceforth, m_1^{RNC} and m_2^{RNC} will denote the first and second moments of the service time distribution using RNC and RR respectively. Note that m_1^{RNC} and m_1^{RR} are the same as $\mathbb{E}[T_{RNC}]$ and $\mathbb{E}[T_{RR}]$. In order to characterize m_2^{RNC} and m_2^{RR}, we will need the distributions of T_{RNC} and T_{RR}. Propositions 5 and 6 characterize m_2^{RNC} and m_2^{RR}. Their proofs are given in [1].

Proposition 5: Let $F(y, K, p)$ denote the cumulative distribution function of a Pascal random variable of order K and

[2]This form for the utility function signifies that users gain a fixed amount of utility from files that meet their delay constraint. This is motivated by real-time applications where users are often not concerned about how long it takes for the file to be transferred as long as the transfer is completed within a certain deadline.

success probability p. Then

$$m_2^{RNC} = \sum_{i=1}^{\infty} i^2 \left[\left(\sum_{\tau=K}^{i} \binom{\tau-1}{K-1} (1-c)^{(\tau-K)} c^K \right)^N - \left(\sum_{\tau=K}^{i-1} \binom{\tau-1}{K-1} (1-c)^{(\tau-K)} c^K \right)^N \right].$$

■

Since our goal is to characterize the delay gains from network coding as compared to scheduling, it is sufficient to obtain lower bounds for m_1^{RR} and m_2^{RR}. The use of lower bounds for scheduling (if attainable) corresponds to the worst-case scenario, because in practice the queueing delay for scheduling will always be larger than the queueing delay obtained using the lower bounds (leading to better comparative gains from network coding).

A lower bound on m_1^{RR} can be trivially obtained from (1):

$$m_1^{RR} \geq \frac{K}{2} + K \sum_{t=1}^{\infty} \left[1 - (1-(1-c)^t)^{KN}\right]. \qquad (6)$$

In order to obtain a lower bound on m_2^{RR}, we define $\hat{T}_{RR} \triangleq \max_{\substack{1\leq i\leq N \\ 1\leq k\leq K}} KW_i^k$, where W_i^k is a geometric random variable with parameter c. Recalling the definition of T_{RR} from Proposition 1-(B), we can see that $\hat{T}_{RR}$ is stochastically less than or equal to T_{RR}, and hence can serve as a lower bound on T_{RR}.

Let $\hat{m}_2^{RR}$ be the second moment of $\hat{T}_{RR}$. Since $\hat{T}_{RR}$ is a lower bound on T_{RR}, $\hat{m}_2^{RR}$ constitutes a lower bound on m_2^{RR}. We characterize $\hat{m}_2^{RR}$ in Proposition 6.

Proposition 6: Let $F_G(y,p)$ denote the distribution function of a geometric random variable with parameter p. Then,

$$\hat{m}_2^{RR} = K^2 \sum_{i=1}^{\infty} i^2 \left[F_G(i,c)^{KN} - F_G(i-1,c)^{KN}\right],$$

where c is the probability of a successful packet transmission on a channel in a given time slot. ■

We use equation 4 along with the first and second moments of the service times process to observe the effect of the maximum allowable mean delay d_{avg}^{max} on the system capacity, i.e., the number of receivers the system is able to support. The coding window K and the user admission rate λ are held constant.

The characterization of m_2^{RNC} given in Proposition 5, although accurate, is cumbersome, requiring the computation of combinations and sums of large orders. In order to efficiently compute m_2^{RNC}, we use the Pascal-to-Gamma approximation suggested by Guenther [7]. We approximate the Pascal distribution function $F(y,K,p)$ by the Gamma distribution function as follows:

$$F(y,K,p) \approx \mathbb{P}(M,X) = \int_0^X \frac{t^{(M-1)}e^{-t/2}}{2^M\Gamma(M)},$$

where $\mathbb{P}(M,X)$ is the Gamma distribution function with parameters α and β, and $M = \alpha = k(1-p)$ and $X = (2y+1)p$. This approximation of the Pascal distribution by a Gamma distribution is not only accurate but also significantly easier to evaluate.

In all subsequent simulations, delay is measured in time slots, and γ and λ are measured in users per time slot. The value of γ is taken to be 10 users per time slot for the rest of this paper. Figures 2 and 3 show plots of the number of receivers N the system can support against the maximum allowable mean delay d_{avg}^{max} for network coding and scheduling respectively. The coding window was held constant at $K = 20$.

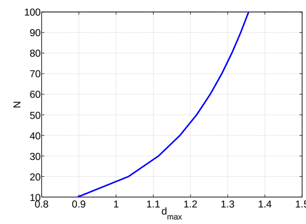

Fig. 2. Number of receivers N against d_{avg}^{max} for random network coding, with $K = 20$

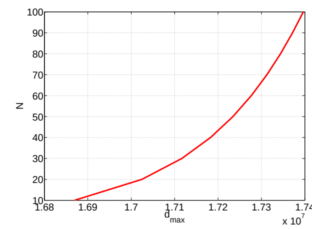

Fig. 3. Number of receivers N against d_{avg}^{max} for round-robin scheduling, with $K = 20$

We observe that with network coding, the system can support a large number of receivers ($N = 100$) for a significantly smaller delay than scheduling. Moreover, the user-to-delay curve for network coding is much steeper than that for scheduling. The number of receivers rises from about $N = 10$ to $N = 100$ as d_{avg}^{max} increases from 0.9 to 1.3, a change of only 0.4. In the case of scheduling, on the other hand, the number of receivers rises from $N = 10$ to $N = 100$ as d_{avg}^{max} increases from 1.69×10^7 to 1.74×10^7, which is a much larger change of 5×10^6. The reason for the huge difference in the two delays is that the user admission rate λ was fixed to the same value for both network coding and scheduling. A large λ translates to enormous delays in the case of scheduling, which means that in order to maintain this value of λ, users must be willing to face inordinately large delays, i.e., their delay constraints must be sufficiently relaxed. Thus, we observe that d_{avg}^{max} in the case of scheduling is considerably large.

Figure 4 shows plots of the user admission rate λ against d_{avg}^{max} for network coding and scheduling. The number of receivers and the file size were held constant at $N = 50$ and $K = 20$ respectively. It is evident that for a given number of receivers and a given coding window, network coding admits a much higher user admission rate than scheduling for a wide range of values of the maximum allowable mean delay. For $K = 20$, the ratio of the user admission rate for network coding to that of scheduling is about six. In other words, over a long duration of time, network coding can on average support six times the number of users that scheduling can support. Figure 5 shows how this ratio varies as the file size changes. For $K = 5$, this ratio is about 3.5 and increases monotonically to approximately 7 at $K = 30$, illustrating that network coding can accommodate significantly more real-time traffic than scheduling.

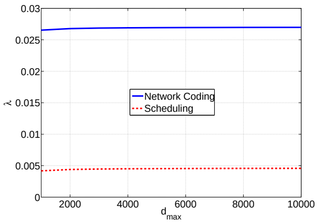

Fig. 4. User admission rate λ against d_{avg}^{max}, $N = 50$, $K = 20$

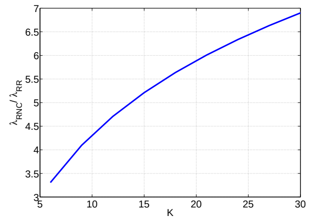

Fig. 5. $\lambda_{RNC}/\lambda_{RR}$ with varying K, for $N = 50$

IV. Extension to General Topologies

In this section, we extend the single-hop cellular downlink model to the multi-hop case, which can be conceived of as a chain of single-hop links placed in series. This approach enables us to analyze multicast settings in general network topologies using results derived for the single-hop case. We analyze general multicast settings via the following two steps:

- Rearrange the general topology into a layered topology.
- Analyze the layered topology as a chain of single-hop networks.

A. An Example of the Multi-hop Case

Consider the multicast setting shown in Figure 6 consisting of two sink nodes, a single source node and some intermediate nodes. Such a setting can be representative of a peer-to-peer network with the source as a central server and the terminals as peers requesting information from the server.

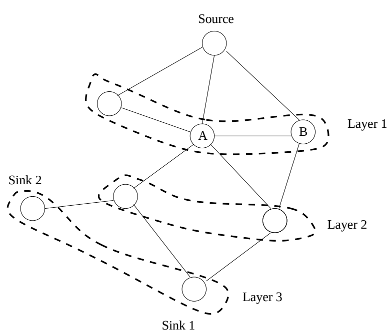

Fig. 6. A multicast setting in a general network topology

The first step is to rearrange the network in a layered topology. Each layer in the layered topology consists of all nodes which can be reached from the source in a given number of hops, i.e., the i^{th} layer consists of nodes that can be reached from the source in i hops (see Figure 6). We assume there is no communication among nodes within the same layer, i.e., we drop all links among nodes within the same layer. For instance, node A in Figure 6 can be reached from the source either in one hop, or in two hops via node B. We define the distance of a node to be the minimum number of hops in which it can be reached from the source. Therefore, both node A and node B are placed in Layer 1, and the link between A and B is dropped. 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. We assume that nodes in the same layer compete for the same network resources and can interfere with one another, while nodes in different layers operate in orthogonal channels (e.g., disjoint frequency bands or time slots) and cannot interfere with one another. The channel separation can be achieved through a simple time or frequency sharing mechanism and allows us to focus exclusively on transmissions between adjacent layers.

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. We assume that transmission succeeds only if a receiver receives exactly one packet in a time slot (no collision). Otherwise transmission fails. 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.

B. Multiple-transmitter Multiple-receiver Systems

Consider a system 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 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 will be connected to a different number of transmitters. The channel conditions on each link are identical to the channel conditions for the single-transmitter multiple-receiver case. Figure 7 illustrates the system topology for $N_s = 3$ and $N_r = 4$. Here, each receiver is connected to two transmitters.

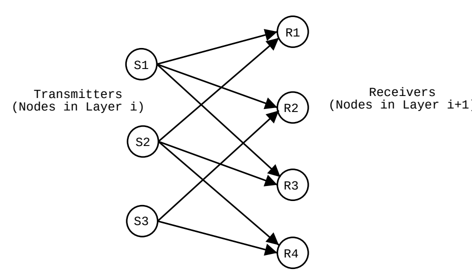

Fig. 7. A multiple-transmitter multiple-receiver system with three transmitters and four receivers

Initially, all of the transmitters possess a single 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 communication among transmitters, a better strategy is for transmitter S_i to attempt transmission with probability p_i in every time slot. For simplicity, we restrict our attention to the symmetric case in which $p_i = p$ for all transmitters. The channel between S_i and, say, receiver R_i is ON with probability c. The probability that R_i successfully receives a packet from S_i is therefore pc. Suppose each receiver is connected to L nodes ($L < N$). Then, 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 $\mathbb{P}(X = 1) = Lpc(1 - pc)^{L-1}$.

In order to maximize throughput, we maximize the probability of a successful transmission given by the expression above. This expression is identical to the probability of a successful capture in the Aloha system. Assuming that the number of transmitters L is reasonably large, say $L \geq 10$, we can use our knowledge of Aloha to conclude that the above probability is maximized when $p = 1/L$, attaining a maximum value of $1/e$. Therefore, the number of packets a receiver receives in one time slot is Bernoulli distributed with a success probability of $1/e$. The mean file completion times for network coding and scheduling are now given by Proposition 1-(C) and (D) respectively, with $c = 1/e$ and $N = N_r$.

Figure 8 illustrates the delay performance of network coding versus scheduling for $K = 30$. There is approximately a four-fold gain in delay from network coding for $N_r = 10$, which increases to five-fold for $N_r = 40$. Indeed, during the course of packet transmission from one layer to the next, we expect the gains from network coding to accumulate from layer to layer. If the depth of the network is large, the cumulative gains from network coding will be significantly higher.

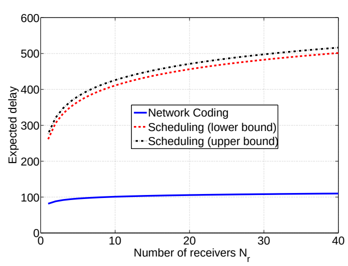

Fig. 8. Delay Performance of Network Coding versus Scheduling (Round robin) for $K = 30$

V. Conclusions

In this work, we have studied scaling laws governing the delay performance of network coding and scheduling in a simple cellular downlink model and have described an extension to general multi-hop topologies where gains from network coding can be realized.

We have analyzed the system for two cases: elastic traffic and inelastic traffic. In the case of elastic traffic, we have shown that the gains from network coding scale directly as the reciprocal of the file size K, and significantly large gains can be realized for large file sizes. In the case of inelastic traffic, we have investigated the performance of network coding under different scaling laws involving the number of receivers N, delay constraint d_{avg}^{max}, throughput λ, and file size K, and have shown that for the same delay constraint, network coding is not only able to support an appreciably larger number of receivers than scheduling but also permits a higher throughput. We have also proposed an extension of the single-hop setting to multi-hop topologies with multiple sources and multiple receivers, and have shown that network coding yields significant delay gains in the multi-hop case as well.

The possibilities for future work are vast. An important extension to the cellular downlink model would be to relax the assumption of Poisson arrivals and study the system under more general arrival processes. Other extensions to the multi-hop case are finding the optimal scheduling strategy for each layer, relaxing the assumption that nodes within the same layer cannot communicate (i.e., allowing for the possibility of links between two nodes in the same layer), relaxing the assumption of synchronization (i.e., a node can start transmitting the file without having to wait for all nodes in the same layer to receive the file), and analyzing multiple multicast (i.e., permitting multiple flows across one link).

References

[1] E. Ahmed. Delay gains from network coding in wireless networks. M.Eng. thesis, Massachusetts Institute of Technology, May 2007.

[2] E. Ahmed, A. Eryilmaz, A. Ozdaglar, and M. Médard. Economic aspects of network coding. In *Proceedings of the 44th Annual Allerton Conference on Communications, Control and Computing*, Monticello, IL, September 2006.

[3] 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.

[4] A. Eryilmaz, A. Ozdaglar, and E. Modiano. Polynomial complexity algorithms for full utilization of multi-hop wireless networks. In *Proceedings of IEEE Infocom*, Anchorage, AK, May 2007.

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

[6] P. Grabner and H. Prodinger. Maximum statistics of *n* random variables distributed by the negative binomial distribution. *Combinatorics, Probability and Computing*, 6(4):179–183, 1997.

[7] W. C. Guenther. A simple approximation to the negative binomial (and regular binomial). *Technometrics*, 14(2):385–389, 1972.

[8] J. F. C. Kingman. Some inequalities for the queue GI/G/1. *Biometrika*, 49:315–324, 1962.

[9] X. Lin and N. Shroff. The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks. In *Proceedings of IEEE Infocom*, Miami, FL, March 2005.

[10] S. Sanghavi, L. Bui, and R. Srikant. Distributed link scheduling with constant overhead, 2007. Technical Report.

[11] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhäuser, Boston, MA, 2004.

[12] X. Wu and R. Srikant. Regulated maximal matching: A distributed scheduling algorithm for multi-hop wireless networks with node-exclusive spectrum sharing. In *Proceedings of* IEEE Conference on Decision and Control, 2005.