STABLE THROUGHPUT FOR MULTICAST WITH INTER-SESSION NETWORK CODING

Randy Cogill
Dept. of Systems and Information Engineering
University of Virginia
rcogill@virginia.edu

Brooke Shrader and Anthony Ephremides
Dept. of Electrical and Computer Engineering
University of Maryland
{bshrader,etony}@umd.edu

***Abstract*— In this work we compare scheduling and coding strategies for a source node serving multiple multicast flows in a network. The coding strategy we consider is a form of random coding proposed in [6] and involves coding across flows, as treated in [2]. We show that there are configurations for which the coding strategy outperforms any scheduling strategy that uses channel state information.**

The model studied in this paper is a multicast version of the downlink model considered in [13]. Fixed-length data packets arrive randomly and independently to a set of multicast flows served by the source node. Each multicast flow is associated with a set of receivers. The channel between the source and each receiver is modeled as an independent packet erasure channel. We first consider the set of scheduling strategies that can observe queue lengths and channel states, and provide an upper bound on the achievable sum stable throughput over all scheduling strategies. We then consider a random linear coding strategy involving coding both within and between the multicast flows and provide a lower bound on the sum stable throughput achievable by this strategy. Our bound on the throughput for the coding strategy accounts for the overhead incurred by random coding. Finally, we show that there are instances for which the coding lower bound is strictly greater than the upper bound for scheduling. Our result provides a simple example demonstrating that the back-pressure algorithm [12], which is optimal for store-and-forward networks, can be inferior to a coding approach for multicast transmission.

This work is an extension of our previous work in [1], where we compared round-robin scheduling to coding, considered coding across (but not within) multicast flows, and disregarded the coding overhead.

I. INTRODUCTION

In this paper we compare scheduling and coding strategies for a source node serving multiple multicast flows in a network. The coding strategy we consider is a form of random coding proposed in [6] and involves coding across flows, as treated in [2]. We show that there are configurations for which the coding strategy outperforms any scheduling strategy that uses channel state information.

The multicast downlink model considered in this paper is shown in Figure 1. In this system, fixed-length data packets arrive randomly and independently at rate λ/K to each of K multicast flows served by a source node. Each multicast flow maintains a queue and is associated with M receivers. The channel between the source and each receiver is modeled as an independent packet erasure channel with erasure probability $1-q$. Packets can only be removed from a queue once they are received by all of the receivers associated with this queue.

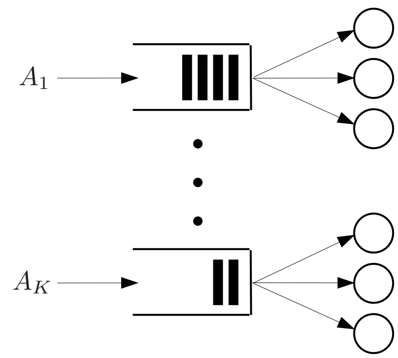

Fig. 1. Multicast downlink model.

We first consider the set of scheduling strategies that can observe queue lengths and channel states, and provide an upper bound on the sum stable throughput over all strategies. In the unicast case, it is well known that the optimal sum stable throughput is achieved by using the Longest Connected Queue policy [13]. The unicast case is depicted in Figure 2. In this case, each queue is connected to its associated receiver with probability q in each time slot. When making a scheduling decision, the transmitter is aware of which queues are connected to their receivers in the current time slot. The Longest Connected Queue policy always transmits a packet from the longest queue connected to a receiver, and sits idle if no queues are connected. This policy is able to stabilize the system for all $\lambda < 1-(1-q)^K$, and no policy is able to stabilize the system for any $\lambda > 1-(1-q)^K$.

The analysis becomes more complex when we attempt to find a similar policy for the multicast case. In each time slot, each queue might be connected to some but not all receivers. It is not clear how to consider the number of connected receivers together with the queue lengths when making a scheduling decision. Also, multicast scheduling is made considerably more complex by the fact that a packet cannot be removed from the system until it has been received by all of its intended receivers. That is, in one time slot a queue might be connected to many receivers, but most of the connected receivers might have already received the packet under consideration in a previous time slot.

978-1-4244-2677-5/08/$25.00

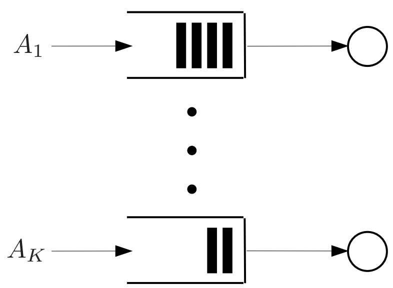

Fig. 2. Unicast downlink model.

In spite of the complexities of multicast scheduling, we show that there is a simple coding strategy that can often achieve higher throughput than any scheduling strategy. We consider a random coding strategy that transmits random linear combinations of the queued packets in each time slot. This coding strategy ignores the connectivities of the queues when encoding packets. So, scheduling can often be outperformed and its complexities can be avoided by using a randomized coding strategy that completely ignores link connectivities and previous packet receptions.

II. Preliminaries

An instance of the multicast downlink model considered in this paper is specified by the parameters K, M, q, and λ. The system consists of K queues, each connected to a set of M distinct receivers. In each time slot, a queue is connected to a specific receiver with probability q. Connectivities are independent across receivers and queues. Each of the queues receive independent Bernoulli arrivals, each with rate λ/K. We will consider the set of sum arrival rates for which this system is stable for two transmission schemes:

(a) **Scheduling with channel state information:** In each time slot, the lengths of the K queues and the connectivities of the MK receivers are observed. Based on this information, one of the K queues is selected and the head-of-line packet from this queue is transmitted. A head-of-line packet is only removed from its queue when it has been received by all M receivers awaiting this packet.

(b) **Random linear coding:** In each time slot, we create a random linear combination of packets in the K queues. Specifically, a coding block of length CK is formed by taking the C packets at the head of each of the K queues. A random linear combination of these CK packets is formed by selecting a random subset of these packets and adding them bitwise. New random linear combinations of these CK packets are transmitted to all MK receivers in each time slot until all MK receivers can decode all CK packets. If a queue contains fewer than C packets when a round of encoding begins, we code over all available packets in this queue. Any arrivals to those queues during transmission will not be included in the current coding block.

One of the key differences between these two schemes is that scheduling only requires that each packet is successfully received by the receivers assigned to it, but random linear coding implicitly requires that each packet is successfully received by every receiver. It seems that this would cause coding to make many unnecessary transmissions. However, we will see that random linear coding still often supports higher sum arrival rates than scheduling.

III. Main Results

The goal of this paper is to show that instances of the multicast downlink model exist that cannot be stabilized by any scheduling policy that uses channel state information, but can be stabilized by coding strategies that do not use channel state information. This implies that coding can stabilize instances that cannot be stabilized a multicast version of the Longest Connected Queue policy. This will be shown by:

- Finding an upper bound on the largest sum arrival rate λ that can be stabilized by a scheduling policy that can observe the queue lengths and link connectivities in each time slot.
- Finding a lower bound on the largest sum arrival rate λ that can be stabilized by a coding strategy that codes across and within queues.
- Showing that instances exist where the coding lower bound is strictly greater than the scheduling upper bound.

A. Stability of scheduling with channel state information

To start, we will consider the throughput achievable by scheduling policies that can observe the queue lengths and link connectivities in each time slot. The following theorem gives an upper bound on the maximum sum arrival rate that can be stabilized by scheduling.

Theorem 1: Let $\mathcal{P}$ be the set of state feedback policies that transmit the head-of-line packet from one of the K queues in each time slot given the current lengths of all queues and the current connectivities of all links. If

$$\lambda > \frac{-K\ln(1-q)}{\ln(M+1)+0.3}, \tag{1}$$

then the queue length process is unstable for all policies $\pi \in \mathcal{P}$.

Proof: Let Ω be the sample space of all joint initial queue lengths, arrival sequences, and link connectivity sequences. Let $Q_i(0)$ be the random variable giving the length of queue i at time 0. Let $A_i(t)$ be the random variable indicating an arrival in queue i at time t. Also, let $I_i^\pi(t)$ be the random variable indicating a departure from queue i at time t under policy π. Since departures from a queue at a given time under a given feedback policy depend only on the initial queue lengths, previous arrivals, and previous connectivities of the links, $I_i^\pi(t)$ depends only on ω. For all ω, i and t define

$$Q_i^\pi(t,\omega) = Q_i(0,\omega) + \sum_{\tau=0}^{t-1}\big(A_i(\tau,\omega) - I_i^\pi(\tau,\omega)\big).$$

Similarly, for each i consider the policy that always attempts to serve queue i. Let $\widehat{I}_i(t)$ be the random variable indicating a departure from queue i at time t under the 'always serve queue i' policy. For all ω, i and t define

$$\widehat{Q}_i(t,\omega) = Q_i(0,\omega) + \sum_{\tau=0}^{t-1}\big(A_i(\tau,\omega) - \widehat{I}_i(\tau,\omega)\big).$$

For all $\omega \in \Omega$ we have

$$\begin{aligned} Q_i^\pi(t,\omega) - \widehat{Q}_i(t,\omega) &= \sum_{\tau=0}^{t-1}\big(\widehat{I}_i(\tau,\omega) - I_i^\pi(\tau,\omega)\big) \\ &\geq 0. \end{aligned}$$

for all π, t and i. Therefore,

$$\sum_{i=1}^K Q_i^\pi(t,\omega) \geq \sum_{i=1}^K \widehat{Q}_i(t,\omega)$$

for all ω, π and t, implying

$$\mathbf{E}\left[\sum_{i=1}^K Q_i^\pi(t)\right] \geq \mathbf{E}\left[\sum_{i=1}^K \widehat{Q}_i(t)\right]$$

for all π and t.

Denote $\lambda_i = \mathbf{E}[A_i(t)]$, where the arrival rate is the same for all t. From Lemma 1 in the Appendix,

$$\liminf_{t\to\infty} \mathbf{E}\left[\sum_{i=1}^K \widehat{Q}_i(t)\right] < \infty$$

only if

$$\lambda_i \leq \frac{-\ln(1-q)}{\ln(M+1)+0.3}$$

for all i. So, if

$$\sum_{i=1}^K \lambda_i > \frac{-K\ln(1-q)}{\ln(M+1)+0.3},$$

then

$$\begin{aligned} \liminf_{t\to\infty} \mathbf{E}\left[\sum_{i=1}^K Q_i^\pi(t)\right] &\geq \liminf_{t\to\infty} \mathbf{E}\left[\sum_{i=1}^K \widehat{Q}_i(t)\right] \\ &= \infty, \end{aligned}$$

for all policies $\pi \in \mathcal{P}$. ■

This bound has a simple interpretation. At any time, the length of a specific queue operating under some scheduling policy cannot be any smaller than the length of this queue if it is served in every time slot. So, we can determine the smallest arrival rate for which a single queue served in every time slot becomes unstable. For any set of arrival rates $\lambda_1, \ldots, \lambda_K$ into the K queues, the system cannot be stabilized by any scheduling policy if any λ_i exceeds this smallest arrival rate.

There is a second interpretation of this bound that can be related to the stability conditions for the unicast system in [13]. To understand this interpretation, consider the unicast downlink operating under a maximum throughput policy. When the total arrival rate into the system is close to the maximum stabilizable rate, all queues are occupied the majority of the time. In most of the time slots where the system is idle, the system is idle because no receivers are connected. This occurs in each time slot with probability $(1-q)^K$. In this case, the system is similar to a single queue with arrival rate $\lambda = \lambda_1 + \cdots + \lambda_K$ transmitting over a single erasure channel with erasure probability $(1-q)^K$. The unicast downlink system is stabilizable only if this single queue system is stable. It is natural to conjecture whether this interpretation can be extended to the multicast system discussed in this paper. Specifically we could ask, "Is it true that stabilizability of the multicast downlink implies stability of a single queue transmitting to M receivers over independent erasure channels each with erasure probability $(1-q)^K$?" If this were true, then a necessary condition for stabilizability could be obtained by applying the result of Lemma 1 in the Appendix to this single queue. It turns out that this would produce the same condition presented in Theorem 1, since the single queue system is unstable if

$$\begin{aligned} \lambda &> \frac{-\ln\big(1-(1-(1-q)^K)\big)}{\ln(M+1)+0.3} \\ &= \frac{-K\ln(1-q)}{\ln(M+1)+0.3}. \end{aligned}$$

B. Stability of coding

We will now consider the coding strategy presented in Section II. Stability of this strategy is closely related to the time required to transmit CK linearly independent combinations of packets to all MK receivers, which we will represent by the random variable T. The following theorem gives an upper bound on the expected value of T.

Theorem 2: Consider the random linear coding strategy described in Section II. For any $CK \geq 16$, the expected number of time slots required to transmit CK linearly independent combinations of packets to all MK receivers satisfies

$$\mathbf{E}[T] \leq \frac{1}{q}\left(CK + \big(\ln(MK) + 0.78\big)\sqrt{CK} + 2.61\right).$$

Proof: Let X_i be the time required to transmit CK linearly independent combinations of the CK encoding packets to receiver i. By Lemma 2 in the Appendix, an upper bound on the expected time for all MK receivers to receive CK linearly independent packets is

$$\begin{aligned} \mathbf{E}[T] &= \mathbf{E}[\max\{X_1, \ldots, X_{MK}\}] \\ &\leq \frac{1}{t}\big(\ln(MK) + \ln(\mathbf{E}[e^{tX_1}])\big). \end{aligned} \tag{2}$$

Each X_i is a sum of CK independent geometric random variables $Y_{0,i}, \ldots, Y_{CK-1,i}$, where $\mathbf{E}[Y_{j,i}] = \frac{1}{(1-2^{j-CK})q}$. So, the right-hand side of (2) is

$$\frac{1}{t}\left(\ln(MK) - \sum_{j=0}^{CK-1} \ln\left(1 - \frac{1-e^{-t}}{(1-2^{j-CK})q}\right)\right)$$

We'll use $t = -\ln(1-qz)$ for some parameter z. By Lemma 3 of the Appendix, $-\ln(1-qz) \geq qz$ for all

$0 \le z < 1$. So, the right hand side of (2) is less than or equal to

$$\frac{1}{qz}\left(\ln(MK) - \sum_{j=0}^{CK-1} \ln\left(1 - \frac{z}{1-2^{j-CK}}\right)\right).$$

Let $\nu_j = z(1-2^{j-CK})^{-1}$. Then

$$-\frac{1}{qz}\sum_{j=0}^{CK-1}\ln\left(1-\frac{z}{1-2^{j-CK}}\right) = -\frac{1}{q}\sum_{j=0}^{CK-1}\frac{\ln(1-\nu_j)}{(1-2^{j-CK})\nu_j}$$

By Lemma 3 in the Appendix, $-\frac{1}{\nu}\ln(1-\nu)$ is convex and $\lim_{\nu\to 0} -\frac{1}{\nu}\ln(1-\nu) = 1$. So, for $0 < \nu \le 1/2$,

$$-\frac{1}{\nu}\ln(1-\nu) \le 1 + \big(4\ln(2) - 2\big)\nu.$$

This gives

$$\begin{aligned}
-\frac{1}{q}&\sum_{j=0}^{CK-1}\frac{\ln(1-\nu_j)}{(1-2^{j-CK})\nu_j} \\
&\le \frac{1}{q}\sum_{j=0}^{CK-1}\frac{1+\big(4\ln(2)-2\big)\nu_j}{1-2^{j-CK}} \\
&= \frac{1}{q}\sum_{j=0}^{CK-1}\left(\frac{1}{1-2^{j-CK}} + z\frac{4\ln(2)-2}{(1-2^{j-CK})^2}\right) \\
&\le \frac{1}{q}\big(CK + 1.61 + (0.78CK + 3.37)z\big)
\end{aligned}$$

if all ν_j satisfy $0 < \nu_j \le 1/2$, which holds if $0 < z \le 1/4$. So, this gives the bound

$$\mathbf{E}[T] \le \frac{1}{q}\left(\frac{\ln(MK)}{z} + (0.78CK + 3.37)z + CK + 1.61\right)$$

for all $0 < z \le 1/4$. If $CK \ge 16$, we can use $z = \frac{1}{\sqrt{CK}}$, giving

$$\mathbf{E}[T] \le \frac{1}{q}\left(CK + \big(\ln(MK) + 0.78\big)\sqrt{CK} + 2.61\right).$$

■

In order for a receiver to decode a set of CK transmitted packets, it must know the set of source packets included in each encoded packet. There are several ways this information can be provided to the receivers. One way to pass this information is to include additional overhead bits in each packet. Assuming that we transmit fixed length packets, including overhead bits in each packet reduces the number of data bits that are sent in each packet transmission, ultimately affecting throughput. We will undertake the analysis of this approach in the next section.

Before analyzing the stability of coding with overhead, we will analyze stability under the assumption that no overhead bits need to be transmitted. This assumption is justified under several circumstances. The subset of packets included in an encoded packet can be determined using a pseudo-random number generator. As discussed in [9], if the transmitter and receivers are synchronized then the receivers can run an identical pseudo-random number generator to the one used at the transmitter. The pseudo-random number generators at the receivers can then be used to identify which source packets are included in each transmitted packet.

Under the assumption that no overhead bits need to be transmitted, the system is stable for all sum arrival rates satisfying

$$\lambda < \frac{CK}{\mathbf{E}[T]}.$$

Using the result of Theorem 2, the system is stable for all

$$\lambda < \frac{qCK}{CK + \big(\ln(MK) + 0.78\big)\sqrt{CK} + 2.61}.$$

Under the assumption of no overhead, the system can be stabilized for all $\lambda < q$, K, and M by choosing C sufficiently large. Clearly, q is the highest arrival rate that can be stably supported by any transmission scheme that does not utilize channel state information.

We conclude this section by showing an instance (specified by λ, q, M, and K) that cannot be stabilized by any scheduling policy that uses channel state information, but can be stabilized by coding without channel state information. Suppose $q = 0.5$, $C = 100$, $K = 2$, and $M = 250$. In this case,

$$\frac{-K\ln(1-q)}{\ln(M+1) + 0.3} \approx 0.238$$

and

$$\frac{qCK}{CK + \big(\ln(MK) + 0.78\big)\sqrt{CK} + 2.61} \approx 0.477.$$

For $\lambda = 0.3$ the coding strategy stabilizes the system, but there is no scheduling strategy that stabilizes the system.

C. Analysis with coding overhead

In the previous section we studied the arrival rates that can be stabilized by coding when transmitted packets are not required to contain any coding overhead. Now we will present a sufficient stability condition for coding across sessions that includes the effect of coding overhead. Recall that each transmitted packet is a linear combination of CK packets. To indicate which packets are present in a particular linear combination, we will require that each encoded packet contains CK bits of overhead.

Let B be the number of bits contained in each packet. All bits of each arriving packet are assumed to be data bits. When coding across CK packets, CK bits of every transmitted packet must be used for coding overhead, and the remaining $B - CK$ bits may contain data. This presents a trade-off when selecting the parameter C used in our coding scheme. Increasing C decreases the average per-packet transmission time, leading to an increased rate that packets can leave the system. However, increasing C also reduces the number of data bits in each packet. The following theorem presents a lower bound on the maximum stable rate using coding with a specific choice of C.

Theorem 3: When including the effect of overhead, coding across all sessions is stable if

$$\lambda < \frac{q(B^{3/4} - B^{1/2} - K)}{B^{3/4} + \big(\ln(MK) + 0.78\big)B^{3/8} + 2.61}. \qquad (3)$$

Proof: We will consider the coding scheme that codes across all sessions and across $C = \lfloor B^{3/4}/K \rfloor$ packets within each session. Within each packet, $CK = K\lfloor B^{3/4}/K \rfloor$ bits are coding overhead and $B - CK$ bits are data. Note that

$$B^{3/4} - K \leq CK \leq B^{3/4}.$$

By Theorem 2, the expected time to transfer $(B - CK)CK$ data bits is less than or equal to

$$\begin{aligned} &\frac{1}{q}\Big(CK + \big(\ln(MK) + 0.78\big)\sqrt{CK} + 2.61\Big) \\ &\qquad \leq \frac{1}{q}\Big(B^{3/4} + \big(\ln(MK) + 0.78\big)B^{3/8} + 2.61\Big). \end{aligned}$$

Also,

$$\begin{aligned} (B - CK)CK &\geq (B - B^{3/4})(B^{3/4} - K) \\ &= (B - B^{3/4} - B^{1/4}K + K)B^{3/4} \\ &\geq (B - B^{3/4} - B^{1/4}K)B^{3/4} \end{aligned}$$

When all queues are loaded, the expected number of data bits leaving the system per unit time is greater than or equal to

$$\frac{q(B - B^{3/4} - B^{1/4}K)B^{3/4}}{B^{3/4} + \big(\ln(MK) + 0.78\big)B^{3/8} + 2.61}$$

If each arriving packet contains B data bits, the expected number of data bits entering the system per unit time is λB. So, the system is stable if

$$\lambda < \frac{q(B^{3/4} - B^{1/2} - K)}{B^{3/4} + \big(\ln(MK) + 0.78\big)B^{3/8} + 2.61}.$$

∎

In the presence of coding overhead, we will show that instances still exist that cannot be stabilized by scheduling but can be stabilized by coding. As an example, suppose $q = 0.5$, $B = 1000$, $K = 2$, and $M = 250$. In this case,

$$\frac{-K\ln(1-q)}{\ln(M+1) + 0.3} \approx 0.238$$

and

$$\frac{q(B^{3/4} - B^{1/2} - K)}{B^{3/4} + \big(\ln(MK) + 0.78\big)B^{3/8} + 2.61} \approx 0.263.$$

For $\lambda = 0.25$ the coding strategy stabilizes the system, but there is no scheduling strategy that stabilizes the system.

Corollary 1: For any given q, K, and B, there exists an M such that (3) is strictly greater than (1) if and only if

$$-\frac{q(B^{3/8} - B^{1/8} - B^{-3/8}K)}{K\ln(1-q)} > 1.$$

Proof: To compare (1) and (3), consider the ratio between the two bounds

$$\begin{aligned} &\left(\frac{q(B^{3/4} - B^{1/2} - K)}{B^{3/4} + \big(\ln(MK) + 0.78\big)B^{3/8} + 2.61}\right)\left(\frac{\ln(M+1) + 0.3}{-K\ln(1-q)}\right) \\ &\qquad = \left(-\frac{q(B^{3/8} - B^{1/8} - B^{-3/8}K)}{K\ln(1-q)}\right)\Psi(M, B, K), \end{aligned}$$

where

$$\Psi(M, B, K) = \left(\frac{\ln(M+1) + 0.3}{B^{3/8} + \ln(M) + \ln(K) + 0.78 + 2.61B^{-3/8}}\right).$$

First suppose that

$$-\frac{q(B^{3/8} - B^{1/8} - B^{-3/8}K)}{K\ln(1-q)} > 1.$$

For any B and K,

$$\lim_{M\to\infty} \Psi(M, B, K) = 1.$$

So, for any B and K there exists M' sufficiently large so that

$$\left(-\frac{q(B^{3/8} - B^{1/8} - B^{-3/8}K)}{K\ln(1-q)}\right)\Psi(M', B, K) > 1.$$

This implies that (3) is strictly greater than (1) for this M'.

Now we'll prove the converse. For all $B \geq 1$ note that $B^{3/8} + 2.61B^{-3/8} \geq 3$. So,

$$\Psi(M, B, K) \leq \frac{\ln(M+1) + 0.3}{\ln(M) + 3.78}$$

It is easy to verify that

$$\frac{\ln(M+1) + 0.3}{\ln(M) + 3.78} \leq 1$$

for all M. So, if

$$-\frac{q(B^{3/8} - B^{1/8} - B^{-3/8}K)}{K\ln(1-q)} \leq 1,$$

then (3) is less than or equal to (1) for all M. ∎

IV. Conclusions

In this paper we compared scheduling and coding strategies for a source node serving multiple multicast flows in a network. We show that there are configurations for which the coding strategy outperforms any scheduling strategy that uses channel state information. To summarize the findings of this paper, we have shown that coding is generally preferable to scheduling when B and M are large and when q and K are small. This makes intuitive sense for the following reasons:

- **The effect of B:** From Theorem 2, the expected number of packets transmitted per unit time is largest when we code across large blocks of packets. However, coding overhead is high when coding across large blocks of packets. For large B, we can code across large blocks of packets and still have the coding overhead be a small fraction of the contents of each packet.
- **The effect of M:** When scheduling, each queue holds its head-of-line packet until it has been obtained by all

receivers associated with this queue. Each time a packet is transmitted, this packet might be received by some receivers that obtained this packet in a previous time slot. Scheduling is difficult with large M since we must either frequently transmit packets that are redundant for some receivers, or postpone transmissions from a queue until only specific receivers are connected. When coding with large block lengths, each transmission is likely to contain information for each receiver. Redundant transmissions only occur when some receivers have received sufficiently many packets to decode, but others are still awaiting packets. For large CK the expected time required to transmit CK packets to the slowest receiver is close to the expected time required to transmit CK packets to any receiver. So, regardless of the value of M, the expected fraction of redundant transmissions can be made arbitrarily small.

- **The effect of q:** When $q = 1$, every transmission reaches all of its intended receivers. For each transmission, the expected number of packets removed from the system is one. Scheduling can stably support all $\lambda < 1$ in this case. However, due to the overhead introduced by coding, as well as the fact that receivers must receive linearly independent combinations, coding can only stably support arrival rates bounded away from one. For example, if 5% of each packet is coding overhead, coding cannot support arrival rates greater than 0.95 even when $q = 1$. So, the benefits of coding become apparent only for smaller q.
- **The effect of K:** Scheduling can take advantage of the diversity gain created by large K. That is, if K is large then it is likely that in each time slot at least one session has many receivers in a favorable state. For coding, large K leads to increased coding overhead when we employ coding across all sessions. If K is large, Corollary 1 shows that coding can only be preferable if the packet size is large enough that the overhead is a small fraction of the contents of each packet.

V. Appendix

The first two lemmas appear in [1], but are repeated here to keep our treatment self-contained.

Lemma 1: Consider a single queue transmitting to M receivers over independent erasure channels each with erasure probability q. The head-of-line packet is transmitted repeatedly and remains in the queue until it is received by all M receivers. This queue is unstable if its arrival rate λ satisfies

$$\lambda > \frac{-\ln(1-q)}{\ln(M+1)+0.3}$$

Proof: We will start by computing a lower bound on $\mathbf{E}[T_2]$, the expected number of time slots required to transmit a packet from this queue. Let Z_i be the number of attempted transmissions to receiver i until it successfully receives its packet. Since the number of attempted transmissions until success for a given receiver is geometrically distributed,

$$\mathbf{P}(Z_i \le n) = 1-(1-q)^n.$$

The time to transmit a packet from the queue is $\max\{Z_1, \ldots, Z_M\}$. Since the channels connecting the transmitter to receivers $1, \ldots, M$ are independent,

$$\begin{aligned}\mathbf{P}(\max\{Z_1,\ldots,Z_M\} \le n) &= \mathbf{P}(Z_1 \le n, \ldots, Z_M \le n)\\ &= \mathbf{P}(Z_1 \le n)\cdots\mathbf{P}(Z_M \le n)\\ &= \big(1-(1-q)^n\big)^M.\end{aligned}$$

So,

$$\begin{aligned}&\mathbf{E}[\max\{Z_1,\ldots,Z_M\}]\\ &\quad= \sum_{n=1}^{\infty} n\mathbf{P}(\max_i\{Z_i\}=n)\\ &\quad= \sum_{n=1}^{\infty} n\big(\mathbf{P}(\max_i\{Z_i\}\le n)-\mathbf{P}(\max_i\{Z_i\}\le n-1)\big)\\ &\quad= \sum_{n=1}^{\infty} n\big(\big(1-(1-q)^n\big)^M-\big(1-(1-q)^{n-1}\big)^M\big).\end{aligned}$$

Letting $\lambda = -\ln(1-q)$,

$$\begin{aligned}&\mathbf{E}[\max\{Z_1,\ldots,Z_M\}]\\ &\quad= \sum_{n=1}^{\infty} n\big(\big(1-e^{-\lambda n}\big)^M-\big(1-e^{-\lambda(n-1)}\big)^M\big)\\ &\quad= \sum_{n=1}^{\infty} n\int_{n-1}^{n}\frac{d}{dx}\big(1-e^{-\lambda x}\big)^M dx\\ &\quad= \lambda M\sum_{n=1}^{\infty} n\left(\int_{n-1}^{n}\big(1-e^{-\lambda x}\big)^{M-1}e^{-\lambda x}dx\right)\\ &\quad\ge \lambda M\int_0^{\infty} x\big(1-e^{-\lambda x}\big)^{M-1}e^{-\lambda x}dx.\end{aligned}$$

The inequality above follows from Hölder's inequality. As shown in [11], the last integral above can be evaluated to give

$$\begin{aligned}&\lambda M\int_0^{\infty} x\big(1-e^{-\lambda x}\big)^{M-1}e^{-\lambda x}dx\\ &\quad= \frac{1}{\lambda}\sum_{i=1}^{M}\frac{1}{i}\\ &\quad\ge \frac{1}{\lambda}+\frac{1}{\lambda}\sum_{i=2}^{M}\int_i^{i+1}\frac{1}{x}dx\\ &\quad= \frac{1}{\lambda}+\frac{1}{\lambda}\int_2^{M+1}\frac{1}{x}dx\\ &\quad= \frac{\ln(M+1)+1-\ln(2)}{-\ln(1-q)}\\ &\quad\ge \frac{\ln(M+1)+0.3}{-\ln(1-q)}.\end{aligned}$$

The first inequality above follows from the fact that $\frac{1}{x}$ is decreasing. So,

$$\mathbf{E}[T_2] \ge \frac{\ln(M+1)+0.3}{-\ln(1-q)}.$$

So, this queue is unstable if

$$\begin{aligned} \lambda &> \frac{-\ln(1-q)}{\ln(M+1)+0.3} \\ &\geq \frac{1}{\mathbf{E}[T_2]} \end{aligned}$$

■

Lemma 2: Suppose $X_1, \ldots, X_N$ are identically distributed, but not necessarily independent random variables. For any $t > 0$,

$$\mathbf{E}[\max\{X_1, \ldots, X_N\}] \leq \frac{1}{t}\big(\ln(N) + \ln\big(\mathbf{E}[e^{tX_1}]\big)\big).$$

Proof: For any $x_1, \ldots, x_N$ and $t > 0$,

$$\begin{aligned} \max\{x_1, \ldots, x_N\} &= \frac{1}{t}\ln\left(\max\{e^{tx_1}, \ldots, e^{tx_N}\}\right) \\ &\leq \frac{1}{t}\ln\left(e^{tx_1} + \cdots + e^{tx_N}\right). \end{aligned}$$

Therefore, for the random variables $X_1, \ldots, X_N$ we have

$$\begin{aligned} &\mathbf{E}[\max\{X_1, \ldots, X_N\}] \\ &\quad\leq \frac{1}{t}\mathbf{E}\left[\ln\left(e^{tX_1} + \cdots + e^{tX_N}\right)\right] \\ &\quad\leq \frac{1}{t}\ln\left(\mathbf{E}[e^{tX_1}] + \cdots + \mathbf{E}[e^{tX_N}]\right) \\ &\quad= \frac{1}{t}\ln\left(N\mathbf{E}[e^{tX_1}]\right) \\ &\quad= \frac{1}{t}\big(\ln(N) + \ln\big(\mathbf{E}[e^{tX_1}]\big)\big), \end{aligned}$$

where the second inequality follows from Jensen's inequality. ■

The following lemma is used in the proof of Theorem 2.

Lemma 3: The function $-\frac{1}{\nu}\ln(1-\nu)$ satisfies the following:

(i) $\lim_{\nu\to 0} -\frac{1}{\nu}\ln(1-\nu) = 1$
(ii) $-\frac{1}{\nu}\ln(1-\nu) \geq 1$ for all $\nu \in (0,1)$
(iii) $-\frac{1}{\nu}\ln(1-\nu)$ is convex on $(0,1)$

Proof: We will show all parts by using the Taylor series expansion for
$-\ln(1-\nu)$ on $[0,1)$:

$$-\ln(1-\nu) = \sum_{i=1}^{\infty}\frac{\nu^i}{i}.$$

To show part (i), note that

$$-\frac{1}{\nu}\ln(1-\nu) = \sum_{i=1}^{\infty}\frac{\nu^{i-1}}{i}.$$

So, it is clear that

$$\lim_{\nu\to 0}\left(1 + \sum_{i=2}^{\infty}\frac{\nu^{i-1}}{i}\right) = 1.$$

To show part (ii), note that all terms of the series for $-\frac{1}{\nu}\ln(1-\nu)$ are non-negative for $\nu \in (0,1)$. So, for all $\nu \in (0,1)$ we have

$$\begin{aligned} -\frac{1}{\nu}\ln(1-\nu) &= 1 + \sum_{i=2}^{\infty}\frac{\nu^{i-1}}{i} \\ &\geq 1. \end{aligned}$$

Finally, to show part (iii), note that

$$\begin{aligned} -\frac{d}{d\nu}\frac{1}{\nu}\ln(1-\nu) &= \frac{d}{d\nu}\sum_{i=1}^{\infty}\frac{\nu^{i-1}}{i} \\ &= \sum_{i=2}^{\infty}\frac{i-1}{i}\nu^{i-2}. \end{aligned}$$

Since all terms of this sum have positive coefficients, it is increasing with increasing ν. Hence, by the first-order conditions for convexity, $\frac{1}{\nu}\ln(1-\nu)$ is convex. ■

REFERENCES

[1] R. Cogill, B. Shrader, and A. Ephremides. Stability analysis of random linear coding across multicast sessions. In *To appear in the 2008 IEEE International Symposium on Information Theory*, 2008.

[2] A. Eryilmaz and D.S. Lun. Control for inter-session network coding. *Submitted to IEEE Transactions on Information Theory*, 2007.

[3] A. Eryilmaz, A. Ozdaglar, M. Medard, and E. Ahmed. On the delay and throughput gains of coding in unreliable networks. *Submitted to IEEE Transactions on Information Theory*, 2007.

[4] L. Georgiadis, M. Neely, and L. Tassiulas. *Resource allocation and cross-layer control in wireless networks*. Foundations and Trends in Networking, Now Publishers Inc., 2006.

[5] T. Ho, Y.-H. Chang, and K. Han. On constructive network coding for multiple unicasts. In *Forty-fourth Allerton Conference on Communication, Control, and Computing*, 2006.

[6] T. Ho, R. Koetter, M. Medard, D.R. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *IEEE International Symposium on Information Theory*, 2003.

[7] S.-Y.R. Li, R.W. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49, 2003.

[8] D.S. Lun, P. Pakzad, C. Fragouli, M. Medard, and R. Koetter. An analysis of finite-memory random linear coding on packet streams. In *Fourth International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt06)*, 2006.

[9] D.J.C. MacKay. Fountain codes. In *Fourth IEE Workshop on Discrete Event Systems (WODES98)*, 1998.

[10] S. Meyn. Stability and asymptotic optimality of generalized MaxWeight policies. *Submitted to SIAM Journal on Control and Optimization*, 2007.

[11] S. Ross. *Introduction to Probability Models*. Academic Press, 2007.

[12] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 1992.

[13] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information Theory*, 1993.