Multicast Queueing Delay: Performance Limits and Order-Optimality of Random Linear Coding

Randy Cogill, Member, IEEE, and Brooke Shrader, Member, IEEE

Abstract—In this work we analyze the average queue backlog at a source node serving a single multicast flow consisting of \( M \) destination nodes. In the model we consider, the channel between the source node and each receiver is an independent identically distributed packet erasure channel. We first develop a lower bound on the average queue backlog achievable by any transmission strategy; our bound indicates that the queue size must scale as at least \( \Omega(\ln(M)) \). We then analyze the queue backlog for a strategy in which random linear coding is performed over groups of packets in the queue; this strategy is an instance of the random linear network coding strategy introduced in [11]. We develop an upper bound on the average queue backlog for the packet-coding strategy to show that the queue size for this strategy scales as \( O(\ln(M)) \). Our results demonstrate that in terms of the queue backlog, the packet coding strategy is order-optimal with respect to the number of receivers.

I. INTRODUCTION

In this paper we analyze the size of the queue backlog for a scheme in which packets are communicated from source to destination nodes by sending algebraic combinations of a batch of packets. This random linear coding scheme was introduced in [11] and shown to provide improved throughput over uncoded transmission for multicast traffic. While random linear coding has been shown to increase multicast throughput in many works, the performance in terms of queue backlog is not fully understood. In this work we provide an upper bound on the average queue backlog for random linear coding and show that as the number of receivers scales up, random linear coding performs as well as any other multicast transmission strategy.

A number of recent works deal with queueing analysis for random linear coding schemes. The work in [12] considers finite-capacity buffers and presents numerical results on the queue blocking probability and delay. A bulk-service queueing model for random linear coding is developed in [15] and numerical results on queueing delay are also presented. The work in [18] proposes a new packet acknowledgment strategy for random linear coding based on acknowledging degrees of freedom; a queue backlog analysis for the policy is provided and the queue size is shown to grow more slowly with load factor than a baseline acknowledgment strategy. The work in [7] provides analytical bounds on the completion time and stable throughput for random linear coding across multiple multicast sessions; the results indicate that although coding across sessions requires unintended recipients to decode packets, the coding strategy can provide larger throughput than uncoded transmission. Recent work in [1] shows that while significant delay penalties are incurred for synchronized coding across flows, asynchronous coding of packets across flows can reduce the queueing delay. Our work differs from previous work in that we present analytical bounds on the average queue backlog and show that random linear coding is order-optimal with respect to the number of receivers.

II. THE PROBLEM

In this paper we consider a problem involving multicast transmission of data packets from a single transmitter. We model the transmitter as a single queue, where packets arrive in this queue according to a Bernoulli process with rate \( \lambda \). In each time slot, a packet is transmitted to all of the \( M \) receivers. Each receiver receives the transmitted packet independently with probability \( q \), independent of past receptions. This system is depicted in Figure 1. Our goal is to devise a transmission scheme that minimizes the expected number of packets in the queue in the steady state.

![Multicast queueing system](image)

\( \lambda \rightarrow \) system

The problem of minimizing expected queue length is deceptively difficult. To see why this is the case, first consider the simple scheme that retransmits the head of line packet until it has been received by all \( M \) receivers. As shown in [7], the expected number of time slots that a packet stays at the head of the queue is greater than or equal to

\[
\frac{\ln(M+1) + 0.3}{-\ln(1-q)}.
\]

(1)

Therefore, this scheme cannot stabilize arrival rates satisfying

\[
\lambda > \frac{-\ln(1-q)}{\ln(M+1) + 0.3}.
\]

(2)
This means that the expected queue length becomes arbitrarily large for arrival rates approaching some value less than or equal to the right-hand side of (2). In other words, for any \( \lambda \), one can choose \( M \) sufficiently large so that the expected queue length under the retransmission strategy is arbitrarily far from optimal.

Another approach is to view the problem of minimizing expected queue length as a purely control theoretic problem. Rather than simply transmitting the head of line packet, in each time slot any packet in the queue can be transmitted. When the state of each channel connecting the transmitter to each receiver is known before transmitting a packet, a controller would use this information, together with the reception history of each packet in the queue to decide which packet to send next. However the complexity and information requirements of such a scheme are clearly very high. On the other hand, without channel state information, throughput of such a scheme is no better than the throughput of the simple retransmission scheme. To see why this is true, if the system is ignorant of the channel states before transmitting a packet, the expected number of times a packet must be transmitted before successful reception still must be greater than or equal to (1), even if transmissions are attempted out of order.

It is well known that the queue can be stabilized for all arrival rates satisfying \( \lambda < q \) using simple random linear coding schemes (see [8], for example). These schemes generally operate by collecting large blocks of packets in the queue, then transmitting encoded packets formed from this block until all receivers can decode all packets in the block. To stabilize rates \( \lambda \) approaching \( q \), arbitrarily large blocks of packets must be formed. The block coding operation must create a backlog that grows with the size of the block. So, it seems that schemes that code over fixed-length blocks of packets are not well suited for the problem of minimizing expected backlog.

III. MAIN RESULTS

In the previous section, we argued that the problem of minimizing expected steady state queue length for multicast is surprisingly complex. In this paper we establish two results for this problem:

- We show that the expected steady-state queue length of any strategy must satisfy
  \[
  \lim \inf_{t \to \infty} \mathbb{E}[Q(t)] \geq \frac{3}{4} \left( \frac{\lambda \ln(M) - 1}{-\ln(1 - q)} \right),
  \]
  where \( Q(t) \) is the number of packets in the queue at time \( t \). This is true even for strategies that exploit channel state information.

- We show the queue length process of a simple random linear coding strategy (at packet departure times) satisfies
  \[
  \lim \sup_{n \to \infty} \frac{1}{n+1} \sum_{k=0}^{n} \mathbb{E}[Q(t_k)] \leq 4 \ln(M) \left( \frac{\lambda}{q - \lambda} \right)^2 + \left( 8 \sqrt{\ln(M)} + 6 \right) \frac{\lambda}{q - \lambda}.
  \]

Here, \( t_n \) is the time at which the \( n \)-th block of packets departs the system.

So, if the queue length process at departure times corresponds to the true queue length process, the expected queue length of the random linear coding strategy is order optimal with respect to the number of receivers.

A. Lower bounds on achievable backlog for multicast

Here we will present a lower bound on the minimum achievable steady state expected queue length for multicast. Specifically, we show that backlog must scale at least logarithmically with \( M \), the number of receivers. In Section III-C, we show that the backlog of the code over queue contents strategy scales logarithmically with \( M \). This implies that coding over the queue contents is order-optimal with respect to the number of receivers in the multicast model. The theorem that will be proved in this section is the following:

**Theorem 1:** Under any strategy
\[
\lim \inf_{t \to \infty} \mathbb{E}[Q(t)] \geq \frac{3}{4} \left( \frac{\lambda \ln(M) - 1}{-\ln(1 - q)} \right).
\]

The proof of this theorem is at the end of this section, after several supporting lemmas are established.

The lower bound presented in this section makes very few assumptions on the strategy used. We will start by stating these assumptions in words, then give a more precise condition that must be satisfied by all policies.

To make the motivation for our assumptions clear, first consider a system composed of a single transmitter and a single receiver. Packets arrive at the transmitter according to a Bernoulli process with rate \( \lambda \). In each time slot the transmitter is connected to the receiver with probability \( q \), independent of past connections.

Various strategies could be used to transmit packets to the receiver. Regardless of the strategy used, we always assume that the system possesses several properties:

1. At any time, the total number of packets that have been removed from the queue does not exceed the total number of packets that have entered the queue.
2. At any time, the total number of packets that have been removed from the queue does not exceed the total number of time slots where the transmitter has been connected to the receiver. So, even if coding is applied, we do not consider schemes that compress packets. We can only transmit \( m \) packets to the receiver if the transmitter and receiver have been connected in at least \( m \) time slots.
3. Connections cannot be used to transmit future arrivals. In other words, the strategy must be causal. We cannot send information about a packet that has not yet arrived in the queue.
(4) The queue starts out empty. Since we are concerned with steady-state queue length, this is without loss of generality.

To make these conditions precise, we’ll introduce some notation. Let \( Q(t) \) be the length of the queue at time \( t \). Let \( A(t) \) be the random variable with \( A(t) = 1 \) if there is an arrival at time slot \( t \) and \( A(t) = 0 \) otherwise. Let \( S(t) \) be the random variable with \( S(t) = 1 \) if the receiver is connected at time slot \( t \) and \( S(t) = 0 \) otherwise. Finally, let \( D(t) \) be the number of departures from the queue at time \( t \).

In terms of \( A, D, \) and \( S \), properties (1), (2), and (3) are captured by the following condition. For all \( \tau \leq t \), we require that \( D \) satisfies

\[
\sum_{k=0}^{t-1} D(k) \leq \sum_{k=0}^{\tau-1} A(k) + \sum_{k=\tau}^{t-1} S(k). \tag{3}
\]

Properties (1) and (2) imply this condition for \( \tau = t \) and \( \tau = 0 \), respectively. Showing that property (3) implies this condition for all other \( \tau \) is a little less obvious. To show this, consider the set of packets that arrived in the system during the interval \([0, t-1]\). We can partition this set into two sets \( A_1 \) and \( A_2 \), where \( A_1 \) is the set of packets arriving in the interval \([0, \tau-1]\) and \( A_2 \) is the set of packets arriving in the interval \([\tau, t-1]\). The total number of departures in the interval \([0, t-1]\) equals the total number of packets from \( A_1 \) that depart the system in the interval \([0, t-1]\) plus the total number of packets from \( A_2 \) that depart the system in the interval \([0, t-1]\). The total number of departures from \( A_1 \) is, by property (1), less than or equal to the total number of packets in \( A_1 \), which is

\[
\sum_{k=0}^{\tau-1} A(k).
\]

Since all packets in \( A_2 \) arrived on or after \( \tau \), property (3) implies that no connection prior to \( \tau \) can be used to serve a packet in \( A_2 \). This, together with property (2), implies that the total number of departures from \( A_2 \) is less than or equal to the total number of connections in the interval \([\tau, t-1]\), which is

\[
\sum_{k=\tau}^{t-1} D(k).
\]

Finally, summing these upper bounds on the number of departures from \( A_1 \) and \( A_2 \) gives (3).

Using property (3), we have the following lemma. This Lemma is very similar to a well known result that obtains a queue length process by applying a reflection mapping to a netput process (see Proposition 6.3 in [2], for example).

**Lemma 1:** If \( Q(0) = 0 \), the queue length at time \( t \) satisfies

\[
Q(t) \geq \max_{t+1 \geq \tau \geq 0} \left\{ \sum_{k=\tau}^{t-1} (A(k) - S(k)) \right\}
\]

for any service policy with departures satisfying (3). Moreover, this inequality is tight when packets depart the queue as services occur.

**Proof:** At any time \( t \),

\[
Q(t) = \sum_{k=0}^{t-1} (A(k) - D(k))
\]

Since

\[
\sum_{k=0}^{t-1} D(k) \leq \sum_{k=0}^{\tau-1} A(k) + \sum_{k=\tau}^{t-1} S(k)
\]

for all \( \tau \in [0, t] \), the queue length satisfies

\[
Q(t) \geq \sum_{k=0}^{\tau-1} A(k) + \sum_{k=\tau}^{t-1} S(k) - \sum_{k=0}^{t-1} S(k)
\]

for all \( \tau \in [0, t] \). Since this holds for all \( \tau \), clearly

\[
Q(t) \geq \max_{t+1 \geq \tau \geq 0} \left\{ \sum_{k=\tau}^{t-1} (A(k) - S(k)) \right\}
\]

The fact that the inequality is tight if packets depart the queue as services occur can be shown by induction. Since we start with \( Q(0) = 0 \),

\[
Q(1) = \max\{0, A(0) - S(0)\} = \max_{0 \leq \tau \leq 0} \left\{ 0, \sum_{k=\tau}^{0} (A(k) - S(k)) \right\}.
\]

Now suppose \( Q(t) \) satisfies

\[
Q(t) = \max_{t+1 \geq \tau \geq 0} \left\{ \sum_{k=\tau}^{t-1} (A(k) - S(k)) \right\}.
\]

Then the queue length at time \( t + 1 \) is

\[
Q(t + 1) = \max\{0, Q(t) + A(t) - S(t)\}
\]

\[
= \max \left\{ 0, \max_{t+1 \geq \tau \geq 0} \left\{ \sum_{k=\tau}^{t} (A(k) - S(k)) \right\} \right\}
\]

\[
= \max_{t+1 \geq \tau \geq 0} \left\{ \sum_{k=\tau}^{t} (A(k) - S(k)) \right\}.
\]

We will now extend these results to the multicast case. Let \( S_j(t) \) be the random variable with \( S_j(t) = 1 \) if the transmitter is connected to the \( j \)-th receiver at time slot \( t \), and \( S_j(t) = 0 \) otherwise. Let \( R_j(t) \) be the number of packets received by receiver \( j \) at time \( t \). Each \( R_j(t) \) must satisfy the property (3). That is, for all \( \tau \leq t \), the total number of packets received by receiver \( j \) up to time \( t - 1 \) satisfies

\[
\sum_{k=0}^{t-1} R_j(k) \leq \sum_{k=0}^{\tau-1} A(k) + \sum_{k=\tau}^{t-1} S_j(k).
\]
Also, since packets do not leave the queue until they have been received by all receivers, the total number of departures in the interval [0, \( t - 1 \)] satisfies

\[
\sum_{k=0}^{t-1} D(k) = \min_j \left\{ \sum_{k=0}^{t-1} R_j(k) \right\}
\]

Under this condition alone, we can establish a lower bound on the achievable expected queue backlog. The next lemma gives a lower bound in terms of the expected value of the minimum of binomial random variables.

**Lemma 2**: Let \( Q(t) \) be the queue length of the multicast system at time \( t \). Under any policy,

\[
\lim_{t \to \infty} \inf E[Q(t)] \geq \sup_{n \geq 0} \{ \lambda n - f_M(n) \},
\]

where \( f_M(n) \) is the expected minimum of \( M \) independent binomial random variables, each with parameters \( (q, n) \).

**Proof**: The total number of packets in the queue at time \( t \) is

\[
Q(t) = \sum_{k=0}^{t-1} A(k) - \sum_{k=0}^{t-1} D(k).
\]

Note that for all \( t \geq \tau \geq 0 \),

\[
\sum_{k=0}^{t-1} D(k) = \min_j \left\{ \sum_{k=0}^{t-1} R_j(k) \right\}
\]

\[
\leq \sum_{k=0}^{t-1} A(k) + \min_j \left\{ \sum_{k=\tau}^{t-1} S_j(k) \right\}.
\]

By Lemma 1, \( Q(t) \) satisfies

\[
Q(t) \geq \max_{\tau \geq 0} \left\{ \sum_{k=\tau}^{t-1} A(k) - \min_j \left\{ \sum_{k=\tau}^{t-1} S_j(k) \right\} \right\}.
\]

By Jensen’s inequality,

\[
E[Q(t)] \geq E\left[ \max_{\tau \geq 0} \left\{ \sum_{k=\tau}^{t-1} A(k) - \min_j \left\{ \sum_{k=\tau}^{t-1} S_j(k) \right\} \right\} \right]
\]

\[
\geq \max_{\tau \geq 0} \left\{ \lambda (t - \tau) - f_M(t - \tau) \right\}
\]

\[
= \max_{\tau \geq 0} \{ \lambda n - f_M(n) \}
\]

where \( f_M(n) \) is the expected minimum of \( M \) independent binomial random variables, each with parameters \( (q, n) \). Finally,

\[
\lim_{t \to \infty} \inf E[Q(t)] \geq \lim_{\tau \to \infty} \sup_{n \geq 0} \{ \lambda n - f_M(n) \} = \sup_{n \geq 0} \{ \lambda n - f_M(n) \}.
\]

Now we are ready to prove Theorem 1.

**Proof of Theorem 1**: Recall that for any \( n \geq 0 \),

\[
\lim_{t \to \infty} E[Q(t)] \geq \lambda n - f_M(n),
\]

where

\[
f_M(n) = E[\min\{X_1, \ldots, X_M\}],
\]

the expected minimum of \( M \) independent binomial random variables \( X_1, \ldots, X_M \) each with parameters \( (q, n) \).

Let

\[
\hat{n}(M) = \frac{3}{4} \left( \frac{\ln(M)}{-\ln(1-q)} \right).
\]

The proof will proceed by showing that

\[
f_M(\hat{n}(M)) \leq \frac{3}{4} \left( \frac{1}{-\ln(1-q)} \right) (4)
\]

for all \( M \geq 1 \). Therefore, for all \( M \geq 1 \) we will have the lower bound

\[
\lambda \hat{n}(M) - f_M(\hat{n}(M)) \geq \frac{3}{4} \left( \frac{\lambda \ln(M) - 1}{-\ln(1-q)} \right).
\]

Note that the largest value taken by any \( (q, n) \) binomial random variable is \( n \). To show (4), we will use the upper bound

\[
E[\min\{X_1, \ldots, X_M\}] \leq nP(\min\{X_1, \ldots, X_M\} > 0).
\]

It is the case that \( \min\{X_1, \ldots, X_M\} > 0 \) if and only if \( X_i > 0 \) for all \( i \). The event \( X_i > 0 \) occurs with probability \( 1 - (1-q)^n \). Since \( X_1, \ldots, X_M \) are independent,

\[
P(\min\{X_1, \ldots, X_M\} > 0)
\]

\[
= P(X_1 > 0) \cdots P(X_M > 0)
\]

\[
= (1 - (1-q)^n)^M
\]

\[
= (1 - e^{n \ln(1-q)})^M.
\]

Using \( n = \hat{n}(M) \) gives

\[
1 - e^{\hat{n}(M) \ln(1-q)} = 1 - e^{(3/4) \ln(M)}
\]

\[
= 1 - M^{-3/4}
\]

\[
= 1 - M^{1/4}.
\]

Since \( M^{1/4} \leq M \) for \( M \geq 1 \), we can use Lemma ?? in the Appendix to obtain

\[
P(\min\{X_1, \ldots, X_M\} > 0) = \left( 1 - \frac{M^{1/4}}{M} \right)^M \leq e^{-M^{1/4}}.
\]

To bound \( f_M(\hat{n}(M)) \),

\[
f_M(\hat{n}(M)) \leq \hat{n}(M)P(\min\{X_1, \ldots, X_M\} > 0)
\]

\[
\leq \frac{3}{4} \left( \frac{\ln(M)}{-\ln(1-q)} \right) e^{-M^{1/4}}
\]

\[
\leq \frac{3}{4} \left( \frac{1}{-\ln(1-q)} \right) \ln(M)e^{-M^{1/4}}.
\]
Note that
\[
\ln(M)e^{-M^{1/4}} = e^{\ln(\ln(M))}e^{-M^{1/4}} = e^{(\ln(\ln(M))-M^{1/4})}
\]

Since \(x^{1/2} \geq \ln(x)\) for \(x \geq 0\) and both \(x^{1/2}\) and \(\ln(x)\) are monotonically increasing for \(x \geq 0\), \(M^{1/4} \geq \ln(\ln(M))\) for all \(M \geq 1\). Therefore, \(\ln(\ln(M)) - M^{1/4} \leq 0\) for all \(M \geq 1\), or equivalently \(\ln(M)e^{-M^{1/4}} \leq 1\) for all \(M \geq 1\). So,
\[
f_M(\tilde{n}(M)) \leq \frac{3}{4} \left( \frac{1}{-\ln(1-q)} \right)
\]
for all \(M \geq 1\), giving the bound
\[
\liminf_{t \to \infty} E[Q(t)] \geq \lambda \tilde{n}(M) - f_M(\tilde{n}(M)) \geq \frac{3}{4} \left( \frac{\lambda \ln(M) - 1}{-\ln(1-q)} \right).
\]

Another interpretation of the bound presented here relates to the response time of a parallel fork-join queue as described in [3]. The fork-join queue is a model employed for parallel processing systems in which an arriving job or customer is directed to a parallel, independent server and service completion can only take place when all \(M\) servers have completed the task. In [3] results for a continuous-time fork-join queue are presented and it is shown that the average response time (or waiting time in the queue) is \(O(\ln(M))\). The multicast problem we consider in this work might also be modeled as a fork-join queue in which each parallel server represents the process of transmission to one of the \(M\) destination nodes. By Little’s Law, the average queue length is equal to \(\lambda\) times the average response time, so our result that the average queue length scales logarithmically with \(M\) is supported by the previous results on fork-join queues.

B. Lower bounds on achievable backlog for general erasure networks

The model of multicast over erasure links can be generalized to an erasure network. This network is described by an acyclic directed graph \(G = (V, E)\) with source vertex \(s\) and \(M\) terminal vertices \(t_1, \ldots, t_M\). We consider a model where time is slotted. In each time slot, edge \((i,j) \in E\) is connected with probability \(q_{ij}\), independent of other edges and past edge connections. An attempted packet transmission from node \(i\) to node \(j\) is successful in a given time slot if edge \((i,j)\) is connected. In the model we consider here, data packets are produced at the source vertex according to a Bernoulli process with rate \(\lambda\), and each packet must be multicast to all terminal vertices. This model incorporates the broadcast nature of wireless networks, since packets transmitted by a node are heard by all neighboring nodes. However, this model assumes that multiple, distinct packets might be received by a node in a given time slot. Also, here we assume all successful packet transmissions are acknowledged by the receiving nodes, and acknowledgments are sent error-free.

Due to time-varying link connectivity and contention for link access, packets must be queued at network nodes. Various strategies can be used for routing packets through the network, scheduling link access at the nodes, and possibly coding among packets in the network. Under any possible strategy, what is the smallest expected queue backlog we can accumulate at all network nodes? Here we outline a framework for analyzing this limit of achievable queue backlog in erasure networks.

The characterization of erasure network capacity considers the capacity across any cut in the network. A cut is a partition of the vertex set of the network into two complimentary subsets, denoted as \(V_c\) and \(\overline{V_c}\), where \(s \in V_c\) and \(t_i \in \overline{V_c}\) for some terminal vertex \(t_i\). The set of edges contained in a cut is \(E_c = \{(i,j) \in E | i \in V_c, j \in \overline{V_c}\}\). Finally, we let the frontier of the cut be the set of vertices
\[
F_c = \{ i \in V_c | (i,j) \in E_c \text{ for some } j \in \overline{V_c} \}.
\]

The capacity of the cut \(c\) is given by
\[
\sum_{i \in F_c} \left( 1 - \prod_{j \in (i,j) \in E_c} (1 - q_{ij}) \right)
\]
This gives the maximum rate which packets can be transmitted across the cut. At best, packets can be transmitted simultaneously from each vertex on the frontier, leading to the sum in this expression. For each vertex in the frontier, only one edge needs to be connected to transmit a packet in a given time slot, leading to the product in this expression. Since all packets traveling from the source to some terminal must cross this cut, the capacity of the cut gives an upper bound on the source rates that can be stably supported by the network. It is also known that the minimum capacity over all cuts is exactly the maximum source rate that can be stably supported.

Let \(A(k)\) and \(D(k)\) denote the number of packets entering and departing the network in time slot \(k\). The total number of packets queued in the network at time slot \(t\) is
\[
Q(t) = \sum_{k=0}^{t-1} A(k) - \sum_{k=0}^{t-1} D(k).
\]
Let \(S_{(i,j)}(t)\) be a random variable with \(S_{(i,j)}(t) = 1\) if edge \((i,j)\) is connected in time slot \(t\), and \(S_{(i,j)}(t) = 0\) otherwise. We can bound the total number of departures in any interval by considering the connectivity of links crossing any cut. For any cut \(c\) and any times \(t_1 \leq t_2\),
\[
\sum_{k=0}^{t_2-1} D(k) \leq \sum_{k=0}^{t_1-1} A(k) + \sum_{k=t_1}^{t_2-1} \sum_{i \in F_c} \left( 1 - \prod_{j \in (i,j) \in E_c} (1 - S_{(i,j)}(k)) \right)
\]
To understand where this bound comes from, suppose we start with \( Q(0) = 0 \). The total number of departures in the interval \([0, t_2 - 1]\) cannot be any greater than the total number of arrivals in this interval. Of the remaining packets arriving in the interval \([t_1, t_2 - 1]\), the total number of these packets departing the network cannot be greater than the total number of connections crossing any cut in the interval \([t_1, t_2 - 1]\). Summing these bounds for the intervals \([0, t_1 - 1]\) and \([t_1, t_2 - 1]\) gives the bound (5). For a network composed of a single source-terminal pair and a single link, this bound is tight.

Using the bound (5),

\[
Q(t_2) = \sum_{k=0}^{t_2-1} A(k) - \sum_{k=0}^{t_2-1} D(k) \\
\geq \sum_{k=1}^{t_2-1} \left( A(k) - \sum_{i \in F_c} \left( 1 - \prod_{j:(j,j) \in E} (1 - S(i,j)(k)) \right) \right)
\]

Since this holds for all times \( t_1 \) and all cuts \( c \),

\[
Q(t_2) \geq \max_{c \in C} \left\{ \sum_{k=1}^{t_2-1} \left( A(k) - \sum_{i \in F_c} \left( 1 - \prod_{j:(j,j) \in E} (1 - S(i,j)(k)) \right) \right) \right\}
\]

In the case of Bernoulli arrivals and Bernoulli link connectivities, the argument of the maximum is simply a linear combination of binomial random variables. That is,

\[
Z_n = \sum_{k=0}^{t-1} A(k)
\]

is a binomial random variable with parameters \( n \) and \( \lambda \). For each \( i \in F_c \),

\[
Y_{c,i,n} = \sum_{k=0}^{n-1} \left( 1 - \prod_{j:(j,j) \in E} (1 - S(i,j)(k)) \right)
\]

is a binomial random variable with parameters \( n \) and \( 1 - \prod_{j:(j,j) \in E} (1 - q_{ij}) \). Since the arrival and link connectivity processes are memoryless, the joint probability mass function of \( Z_n \) and all of the \( Y_{c,i,n} \) is independent of \( t \).

The steady-state expected backlog is lower bounded as

\[
\liminf_{t \to \infty} \mathbb{E}[Q(t)] \geq \mathbb{E} \left[ \sup_{n \geq 0, c \in C} \left\{ Z_n - \sum_{i \in F_c} Y_{c,i,n} \right\} \right] \quad (6)
\]

The random variables \( Z_n - \sum_{i \in F_c} Y_{c,i,n} \) are fairly straightforward to analyze individually. However, the expectation of their supremum, particularly since these random variables are correlated, is significantly more difficult to analyze. However, as was shown in the previous section, we have obtained analytical results in a special case.

Here we will also show a simple network example where a lower bound can be computed numerically. Consider the network shown in Figure 2. A cut separating the source \( s \) from terminals \( t_1 \) and \( t_2 \) is also shown in the figure. In this example there are seven directed cuts separating the source from each of the terminals. For the case where \( q_{ij} = 0.75 \) for all edges, a curve of the expected steady-state value of this lower bound (computed by taking empirical averages) is shown in Figure 3. This figure gives a lower bound on the steady state expected number of packets queued in the network when packets enter the network at vertex \( s \) according to a Bernoulli processes with rate \( \lambda \).

An interesting observation regarding throughput can be made immediately from the bound (6). By applying Jensen’s inequality to the right-hand side, we see that

\[
\liminf_{t \to \infty} \mathbb{E}[Q(t)] \geq \sup_{n \geq 0, c \in C} \left\{ \lambda n - \sum_{i \in F_c} \left( 1 - \prod_{j:(j,j) \in E} (1 - q_{ij}) \right) n \right\}
\]

The lower bound can be made arbitrarily large, and hence the rate \( \lambda \) cannot be stably supported, if there exists some cut with

\[
\lambda > \sum_{i \in F_c} \left( 1 - \prod_{j:(j,j) \in E} (1 - q_{ij}) \right).
\]

This condition is exactly the necessity part of the capacity theorem for erasure networks [9].

C. A simple strategy and an upper bound

Now we will analyze the behavior of the multicast queue under a simple random linear coding strategy that we call ‘code over queue contents’. This strategy sequentially performs rounds of encoding, each of which lasts several time slots. When a round of encoding begins, all packets in the queue are selected. Let \( C \) be the total number of packets in the queue at the start of a round of encoding. Encoded packets formed from random linear combinations of these packets \( C \) are then sent to the receivers. Any arrivals to the queues during a round of encoding will not be considered until the next round of encoding. The round of encoding ends when all receivers can decode all \( C \) packets. These \( C \) packets are removed from their queue, then the next round of encoding begins.

Each encoded packet is formed by randomly and uniformly selecting coefficients \( a_i \in \{0,1\} \) for \( i = 1, \ldots, C \) and taking a linear combination of the \( C \) head-of-line packets, where the \( i \)th packet in the linear combination is multiplied by \( a_i \). Therefore, each encoded packet is a random linear combination of the \( C \) packets in the current coding block. A receiver can recover the original \( C \) packets once it has received \( C \) linearly independent combinations of encoded packets.
Recall that $Q(t)$ denotes the number of packets in the queue at time $t$. The queue length process $Q(t)$ generally does not evolve as a Markov chain. This is because each round of encoding lasts several time slots, and the distribution of the length of a round of encoding is not memoryless. Although the process $Q(t)$ is not a Markov chain, the process $Q(t_0), Q(t_1), Q(t_2), \ldots$ is a Markov chain, where $t_0, t_1, t_2, \ldots$ are the starting times of successive rounds of encoding. Thus, we can apply tools for Markov chains to analyze the average value of the embedded Markov chain $Q(t_n)$. This will provide the steady-state average value of the queue backlog at the start of each round of encoding.

Before presenting Theorem 2, the main result of this section, we will state a general lemma that is used in its proof. A proof of this lemma can be found in [5].

**Lemma 3:** Let $X(t)$ be a Markov chain with countable state space $\mathcal{X}$. Let $r : \mathcal{X} \rightarrow \mathbb{R}$ be a cost associated with being in each state in $\mathcal{X}$, and let $h : \mathcal{X} \rightarrow \mathbb{R}_+$ be a nonnegative function on $\mathcal{X}$. Then

$$
\limsup_{n \rightarrow \infty} \frac{1}{n+1} \sum_{k=0}^{n} E[r(X(k))] \leq \sup_{x \in \mathcal{X}} \{ r(x) + E[h(X(t+1)) | X(t) = x] - h(x) \}.
$$

**Theorem 2:** The steady-state average of the embedded Markov chain $Q(t_k)$ satisfies

$$
\limsup_{n \rightarrow \infty} \frac{1}{n+1} \sum_{k=0}^{n} E[Q(t_k)] \leq \left[ 4 \ln(M) \left( \frac{\lambda}{q-\lambda} \right)^2 + \left( 8\sqrt{\ln(M)} + 6 \right) \frac{\lambda}{q-\lambda} \right].
$$

**Proof:** Throughout this proof, we will let $\rho = \lambda/q$ denote the load factor associated with the queue. To prove the upper bound, we will use the bound given in Lemma 3. Specifically, we will use the function

$$
h(x) = \frac{2x}{1-\rho}.
$$

By applying Lemma 3 we get

$$
x + E[h(Q(t_{i+1}) | Q(t_i) = x)] - h(x) = x + \frac{2(\lambda E[T(x)] - x}{1-\rho}
$$

where $E[T(x)]$ denotes the expected time to transmit a coding block containing $x$ packets. By Theorem 2 in [8],

$$
E[T(x)] \leq \frac{1}{q} \left( x + 2\sqrt{(0.78x + 3.37) \ln(M)} + 2.61 \right).
$$

Using this bound on $E[T(x)]$, we get the upper bound

$$
x + E[h(Q(t_{i+1}) | Q(t_i) = x)] - h(x) \leq \frac{\rho(2\alpha(x) - 1) - 1}{1-\rho} x
$$

where

$$
\alpha(x) = \frac{x + 2\sqrt{(0.78x + 3.37) \ln(M)} + 2.61}{x}.
$$

The value of $x \geq 0$ that maximizes (7) is

$$
x = 4 \ln(M) \left( \frac{\rho}{1-\rho} \right)^2,
$$

and the associated maximum is

$$
4 \ln(M) \left( \frac{\rho}{1-\rho} \right)^2 + \left( 8\sqrt{\ln(M)} + 6 \right) \frac{\rho}{1-\rho}.
$$

This theorem shows that the average queue length at departure times for the ‘code over queue contents’ strategy scales as $O(\ln(M))$.

**IV. CONCLUSIONS**

In this paper we considered a simple multicast model, where a single transmitter sends packets to $M$ receivers over lossy links. The transmitter is equipped with a queue, and our goal was to find a transmission strategy that minimizes the expected number of packets in the queue. While finding the queue length minimizing strategy is still an open problem, here we found a lower bound on achievable performance and an upper bound on the performance for a random linear coding strategy. Specifically, we have shown that queue length must scale as $\Omega(\ln(M))$, and that queue length under the random linear coding strategy scales as $O(\ln(M))$. Hence, the random linear coding strategy is order-optimal with respect to the number of receivers.

In addition to our analysis for the one-hop multicast model, we provided a framework for analyzing multicast in more general erasure networks. This framework provides a method for lower bounding the minimum achievable total queue backlog throughout a network. Here we have shown an example where a lower bound on achievable backlog was computed numerically for a two-hop network.

**REFERENCES**


