*Online Network Coding for Time-Division Duplexing*

# Online Network Coding for Time-Division Duplexing

Daniel E. Lucani
RLE, MIT
Cambridge, Massachusetts, 02139
Email: dlucani@mit.edu

Muriel Médard
RLE, MIT
Cambridge, Massachusetts, 02139
Email: medard@mit.edu

Milica Stojanovic
Northeastern University
Boston, Massachusetts, 02115
Email: millitsa@mit.edu

***Abstract*— We study an online random linear network coding approach for time division duplexing (TDD) channels under Poisson arrivals. We model the system as a bulk-service queue with variable bulk size and with feedback, i.e., when a set of packets are serviced at a given time, they might be reintroduced to the queue to form part of the next service batch. We show that there is an optimal number of coded data packets that the sender should transmit back-to-back before stopping to wait for an acknowledgement from the receiver. This number depends on the latency, probability of packet erasure, degrees of freedom at the receiver, the size of the coding window, and the arrival rate of the Poisson process. Random network coding is performed across a moving window of packets that depends on the packets in the queue, design constraints on the window size, and the feedback sent from the receiver. We study the mean time between generating a packet at the source and it being "seen", but not necessarily decoded, at the receiver. We also analyze the mean time between a decoding event and the next, defined as the decoding of all the packets that have been previously "seen" and those packets involved in the current window of packets. Inherently, a decoding event implies an in-order decoding of a batch of data packets. We present numerical results illustrating the trade-off between mean delay and mean time between decoding events.**

## I. Introduction

The study of network coding in large latency time division duplexing (TDD) channels, i.e., when nodes can only transmit or receive, but not both at the same time, was first considered in Ref. [1]. This reference studied the problem of transmitting a batch of $M$ data packets through a link using random linear network coding and carefully chosen feedback. The objective was to minimize the mean time to complete transmission of the $M$ data packets by choosing how many coded packets to transmit back-to-back before stopping to listen for an acknowledgement. This scheme was more thoroughly characterized in Ref. [2], [3], and extended to the cases of one-to-all broadcast and all-to-all broadcast in [4] and [5], respectively. Ref. [6] and [7] provide practical considerations in terms of the use of small field sizes and a systematic approach that reduces complexity. The assumption in these references was that the source had $M$ data packets in its buffer before starting transmission. We are interested in studying what happens if there is a random arrival of data packets to the sender.

The problem of queueing for network coding has been considered previously to account for burstiness or losses. Ref. [10] and [11] studied a system with random linear coding, slotted time, and a Bernoulli arrival process. Ref. [8] considered a TDD channel, where the time is not slotted and the service time depends on the size of the bulk of packets being sent. Ref. [8] studied the problem of queueing for the scheme proposed in Ref. [1], i.e., [8] considered the problem of transmitting a batch of packets completely to the receiver and getting an acknowledgement at the sender before starting with the transmission of a new batch. The work of Sundararajan *et al*, e.g.,[9], showed a means to operate network coding in an online manner rather than following a batch by batch approach. The authors used a feedback channel to report the status of the receiver in order to provide efficient queue management at the sender.

We extend the work of Ref. [8] to consider an online network coding approach. This work also provides an extension to the results of [9] in the case of 1) a TDD channel which imposes a limited feedback constraint, 2) the use of a general service time, dependent on the number of packets combined in a given transmission, and 3) Poisson arrivals. Another important contribution of our work is to characterize the time between decoding events, where a decoding event constitutes the decoding at the receiver of all the packets that have been involved in linear combinations up to that moment. A decoding event implies an in-order decoding of a batch of data packets and does not refer to the decoding of a packet of random position in the stream of packets. We provide heuristics for the computation of the optimal number of coded packets to be transmitted before stopping to listen for an acknowledgement. We present numerical results for the mean delay of a packet and the mean time between decoding events for different choices of the arrival rate to illustrate the trade-off between these two metrics.

The paper is organized as follows. Section 2 presents preliminary concepts. In Section 3, we describe the system model and assumptions. In Section 4, we present the queueing model of the system. In Section 5, numerical examples are provided. Conclusions are summarized in Section 6.

## II. Preliminaries

We can think of packets as vectors over a finite field. Since we focus on linear network coding, we can think of the state of knowledge of a node as a vector space over the field. Ref. [9] showed that with a proper use of feedback it is possible to perform network coding in an online manner. The authors relied on acknowledging every new degree of freedom (dof), i.e., referring to a new dimension or independent linear combination of the original data packets at the receiver, that was successfully delivered to a receiver. This reference showed that using the feedback on dof required the queue to store a basis for a coset space with respect to the subspace of knowledge common to all the receivers. The authors defined a specific way of computing this basis using the notion of a node "seeing" a data packet. Let us define this concept before continuing.

**Definition** Index of a packet: For any positive integer $k$, the $k$-th packet that arrives at the sender is said to have index $k$.

**Definition** Seeing a packet: A node is said to have "seen" an original packet $p_k$, with index $k$, if it has received enough information to compute a linear combination of the form $(p_k + q)$, where $q$ is itself a linear combination involving only packets with an index greater than that of $p_k$, i.e., of index greater than $k$. Note that decoding a packet implies seeing that packet, which corresponds to $q = 0$.

In this work, we use a similar feedback scheme to that presented in [9]. However, the TDD constraint on the channel requires us to use feedback more sporadically, i.e., we cannot send an acknowledgement (ACK) packet for every successfully received dof. Instead, we aim to determine the number of coded packets to transmit before the sender stops to listen for an ACK packet. This ACK packet will report the last consecutive "seen" packet, say of index $k$. This allows the sender to remove all packets with index $k$ or less.

Note that we do not guarantee that the seen packets will be decoded immediately, similar to the decode-when-seen algorithm in [9]. In general, there is a delay in decoding the data packets, because the receiver has to collect enough independent linear combinations involving the unknown packets. In particular, we are interested in how often a decoding event occurs, which is defined below.

**Definition** Decoding event: For any positive integer $k$, if random combinations at the sender have involved up to the $k$-th packet, it is said that a decoding event occurs if the receiver decodes the $k$-th packet and all the packets before it after a transmission of a group of coded packets by the sender. In other words, it constitutes the decoding at the receiver of all the packets that were involved in random linear combinations up to that moment.

## III. System Model

We consider a sender that wants to transmit information through a link of data rate $R$ [bps]. The channel is modeled as a packet erasure channel with a propagation delay $T_{prop}$. Nodes can only transmit or receive, but not both at the same time. The sender uses random linear network coding to generate coded data packets. The packets included in a linear combination at a given time are determined by the ACK packets received up to that time, the packets in the queue, and the window of coding.

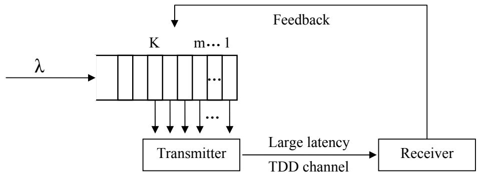

Fig. 1. Bulk queue model with feedback for online network coding for TDD channels.

We consider each data packet to be of fixed-length $n$ bits. The coded packets contain a coded data portion of size, $n$ bits, as well as a header and the random coding coefficients used in the linear combination. Each coefficient is represented by $g = \log_2 q$ bits for encoding over a field of size $q$. The information header is of size $h$ bits. In general, the total number of bits per packet depends on the number of packets combined. For simplicity, we assume that the coded packet contains space for the maximum number of coefficients allowed by the window, i.e., $K$. Therefore, the number of bits in each coded packet is $h + n + gK$. The duration of transmission of a coded packet is then $T_p = \frac{h+n+gK}{R}$.

We consider that the data packets arrive to a source node through a Poisson process with rate $\lambda$ packets/s. Upon arrival, the data packet is placed in a buffer to await encoding and transmission to the receiver, as in Fig. 1. The buffer forms a first-in-first-out (FIFO) queue. However, some of the packets will be fed back to the queue because they were not successfully delivered. That is, the ACK packet from the receiver contains an index number that is lower than the index of some of the packets used in the generation of the previous batch of coded packets.

The size of the coding window of packets is variable, where $m \leq M \leq K$. The pair $(m, K)$ constitutes the range of the bulk size or the size of the coding window used to perform random linear network coding [15]. If the buffer has fewer than $m$ data packets, the system will wait until $m$ packets arrive before providing service. If the buffer contains more than $K$ packets, the system will service exactly $K$ packets. Finally, if the buffer has $M$ packets with $m \leq M \leq K$, then the system will service $M$ packets.

Note that the service time depends on the number of data packets taken from the queue at any time, i.e., the service time distribution is general but it depends on the size of the batch being transmitted. Thus, we can extend the bulk queueing model with feedback developed in [14] to study this problem.

We consider that the sender can transmit coded packets back-to-back before stopping to wait for an ACK packet of duration $T_{ack}$. The waiting time $T_w$ constitutes the time between the sender stopping its transmission and fully receiving the ACK packet, i.e., $T_w = 2T_{prop} + T_{ack}$. The ACK packet feeds back the index of the last consecutive seen data packet. In a sense, this is also providing the number of dof that are still required to decode successfully at the receiver if no new packets are included in future combinations. The number of coded packets sent back-to-back depends on the value of $M$, i.e., the number of packets that will be included in the linear combinations.

Transmission begins after an ACK packet is received and $j \geq m$ packets are in the queue. At this point, $j$ information packets are taken from the queue, which are encoded into $N_j \geq j$ random linear coded packets, and transmitted. The ACK informs the transmitter about the index of the last consecutively seen packet. At this point, the source may have received new data packets. Let's say that $i$ packets will be used to generate linear combinations in the next round of transmission. Then, the transmitter sends $N_i$ coded packets, and so on. If the number of packets in the queue exceeds $K$, then $K$ packets are involved in the linear combinations in the next transmission, and $N_K$ coded packets are transmitted. The time between a transmission of packets and receiving an ACK packet is $t_i = N_i T_p + T_w$ for $i = m, ..., K$, for any $i > K$ the transmission time is $t_K$.

## IV. Queueing Model

The system model discussed in the previous section is very similar to the bulk queueing model with feedback studied in [14]. This bulk queueing model considers Poisson arrivals and a general service time that depends on the bulk size. However, we consider a minimum batch size $m$ that could be different from 1, which was the assumption in [14]. Also, the system model is very similar to the bulk queue model studied in [13] with the addition of packets having some probability of being fed back to the queue. This work generalizes these previous results in order to accurately analyze our proposed system. We use a similar notation to that in [13]. The transition probability of the number of packets in the queue is given by

$$P = \begin{bmatrix} a_0^{(m)} & a_1^{(m)} & \cdots & a_K^{(m)} & a_{K+1}^{(m)} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_0^{(m)} & a_1^{(m)} & \cdots & a_K^{(m)} & a_{K+1}^{(m)} & \cdots \\ a_0^{(m+1)} & a_1^{(m+1)} & \cdots & a_K^{(m+1)} & a_{K+1}^{(m+1)} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_0^{(K)} & a_1^{(K)} & \cdots & a_K^{(K)} & a_{K+1}^{(K)} & \cdots \\ 0 & a_0^{(K)} & \cdots & a_{K-1}^{(K)} & a_K^{(K)} & \cdots \\ 0 & 0 & a_0^{(K)} & \cdots & a_{K-1}^{(K)} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

where $a_k^{(j)}$ is the probability of having $k$ arrivals plus fed back packets during a service of type $j$.

### A. Moment Generating Function of Transition Probabilities

The moment generating function of the transition probabilities can be computed in two ways, with identical results. The first is to consider that the new state of the queue can be computed based on the previous state of the queue, the packet arrivals, and packets taken out of the queue based on the information provided by the ACK packet, i.e., $q_{new} = q_{old} + A(q_{old}) - D(q_{old})$, where $q_{new}$ is the new state of the queue, $q_{old}$ is the previous state of the queue, $A(q_{old})$ and $D(q_{old})$ are the number of packets that arrived to the queue and were taken out it depending on $q_{old}$, respectively. Another perspective is to consider that the new state of the queue depends on the previous state, packet arrivals and packets being fed back to the queue, i.e., $q_{new} = (q_{old} - K)^+ + A(q_{old}) + F(q_{old})$, where $F(q_{old})$ indicates the number of packets being fed back to the queue, given that the previous state is $q_{old}$, and $(X)^+ = \max\{0, X\}$, as in [14]. We choose in this analysis the former representation.

The arrivals are characterized by a Poisson distribution that depends on the previous state of the queue. The distribution for the packets taken out of the queue given that the queue is in state $i$, the set of values of $N_i$'s is $\{N_i\}$, and a link of erasure probability $Pe$, is given by:

$$\mathcal{P}_{(i,\{N_i\},Pe)}(k) = \tag{1}$$
$$\begin{cases} \binom{N_i}{k}\left(\frac{1-Pe}{Pe}\right)^k Pe^{N_i} & \text{if } 0 \leq k < i \text{ and } i \leq K, \\ \sum_{a=i}^{N_i}\binom{N_i}{a}\left(\frac{1-Pe}{Pe}\right)^a Pe^{N_i} & \text{if } k = i \text{ and } i \leq K, \\ \binom{N_K}{k}\left(\frac{1-Pe}{Pe}\right)^k Pe^{N_K} & \text{if } 0 \leq k < i \text{ and } i > K, \\ \sum_{a=K}^{N_K}\binom{N_K}{a}\left(\frac{1-Pe}{Pe}\right)^a Pe^{N_K} & \text{if } k = K \text{ and } i > K, \end{cases}$$

The moment generating function of the departures is given by

$$M_{(i,N_i,Pe)}^{(i)}(s) = \sum_{k=0}^{k=i} e^{sk}\mathcal{P}_{(i,N_i,Pe)}(k) = \tag{2}$$
$$\sum_{k=0}^{i-1}\left(\binom{N_i}{k}\left(\frac{1-Pe}{Pe}\right)^k Pe^{N_i}\left[e^{sk} - e^{si}\right]\right) + e^{si}. \tag{3}$$

Using properties of the moment generating function and the fact that $q_{new} = q_{old} + A(q_{old}) - D(q_{old})$, then, the moment generating function of the transition probabilities when the system is in state $i$ is given by

$$M_T^{(i)}(s) = e^{\lambda t_i}(e^s - 1)e^{si}M_{(i,N_i,Pe)}^{(i)}(-s). \tag{4}$$

### B. Stationary Probabilities

Let us define

$$T^{(j)}(z) = \sum_{k=0}^{\infty} a_k^{(j)} z^k =$$
$$e^{\lambda t_i}(z-1)\left[\sum_{k=0}^{i-1}\left(\binom{N_i}{k}\left(\frac{1-Pe}{Pe}\right)^k Pe^{N_i}\left[z^{i-k} - 1\right]\right) + 1\right].$$

as a z-transform of the transition probabilities, and note that

$$a_k^{(j)} = \frac{1}{k!}\frac{\partial^k}{\partial z^k}T^{(j)}(z)\Big|_{z=0}. \tag{5}$$

Let us denote by $\Pi(z) = \sum_{i=0}^{\infty}\pi_i z^i$ the corresponding generating function of the stationary probabilities. Ref. [13] showed that $\Pi(z)$ can be expressed as

$$\begin{aligned} \Pi(z) &= \frac{T^{(K)}(z)\sum_{i=0}^{K}\pi_i z^i - z^K T^{(m)}(z)\sum_{i=0}^{m}\pi_i}{T^{(K)}(z) - z^K} \\ &\quad - \frac{\sum_{i=m+1}^{K}\pi_i T^{(i)}(z)}{T^{(K)}(z) - z^K} \end{aligned} \tag{6}$$

which provides an expression for $\Pi(z)$ in terms of its first $K+1$ coefficients $\pi_0, ..., \pi_K$. Determining these $K+1$ coefficients provides a full characterization of the stationary probabilities, similar to the result in [13].

Using the same techniques as Ref. [13], we can prove that $T^{(K)}(z) - z^K$ has exactly $K$ zeros satisfying $|z| \leq 1$ assuming that $T^{(K)}(z)$ has a radius of convergence greater than one. Denoting the roots as $1, z_1, ..., z_{K-1}$ and assuming that they are different, we note that the numerator of (6) has to vanish for $z_1, ..., z_{K-1}$ which gives us $K-1$ linear equations

$$\begin{aligned} T^{(K)}(z_k) \textstyle\sum_{i=0}^{K} \pi_i z_k^i - z_k^K T^{(m)}(z_k) \sum_{i=0}^{m} \pi_i \\ - \textstyle\sum_{i=m+1}^{K} \pi_i T^{(i)}(z_k) = 0 \end{aligned} \qquad (7)$$

for $k = 1, ...K-1$. Also, the numerator vanishes trivially for $z = 1$ for both the numerator and the denominator in (6). We thus need one more linear equation. To obtain this we use l'Hôspital's rule to exploit the fact that $\Pi(1) = 1$. This translates to

$$1 = \sum_{i=0}^{m} \left[ 1 + \frac{i - \theta_m}{\theta_K - K} \right] \pi_i + \sum_{i=m+1}^{K} \left[ \frac{\theta_K + i - \theta_i}{\theta_K - K} \right] \pi_i$$

where we have defined $\left.\frac{\partial T^{(i)}(z)}{\partial z}\right|_{z=1} = \theta_i$. The final linear equation to fully characterize $\Pi(z)$ is given by

$$(a_0^m - 1)\pi_0 + a_0^m \pi_1 + \cdots + a_0^m \pi_m + a_0^{m+1} \pi_{m+1} + a_0^K \pi_K = 0.$$

We can use techniques developed by Ref. [16] in order to compute the roots of $T^{(K)}(z) - z^K$. In fact, with a simple solver it is sufficient to find the root of $z\left(T^{(K)}(z)\right)^{-1/K} - e^{2ki\pi/K}$ with $i = \sqrt{-1}$, for every value of $k \in \{0, ..., K-1\}$, which provides us with the $K$ required roots.

### C. Queue of Finite Capacity

The general solution requires the calculation of the roots of $T^{(K)}(z) - z^K$. This can result in numerical inaccuracies and becomes increasingly difficult when the decision variable $K$ assumes a larger value. For these reasons, we simplify the problem considering that the system has a capacity of $B$ packets waiting to be serviced. The transition probability for this case is

$$P = \begin{bmatrix} a_0^{(m)} & a_1^{(m)} & \cdots & a_{B-1}^{(m)} & R(B-1, m) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_0^{(m)} & a_1^{(m)} & \cdots & a_{B-1}^{(m)} & R(B-1, m) \\ a_0^{(m+1)} & a_1^{(m+1)} & \cdots & a_{B-1}^{(m+1)} & R(B-1, m+1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_0^{(K)} & a_1^{(K)} & \cdots & a_{B-1}^{(K)} & R(B-1, K) \\ 0 & a_0^{(K)} & \cdots & a_{B-2}^{(K)} & R(B-2, K) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & a_0^{(K)} & \cdots & a_{B-K}^{(K)} & R(B-K, K) \end{bmatrix}$$

where $R(k, l) = 1 - \sum_{j=0}^{k} a_j^{(l)}$. In order to compute the stationary distribution, it suffices to solve $\bar{\pi} = P\bar{\pi}$, with $\bar{\pi} = [\pi_0, \pi_1, ..., \pi_B]^T$, under the constraint that $\sum_{i=0}^{B} \pi_i = 1$.

### D. Mean Delay

Let us define the mean delay $E[D]$ of a packet as the time that elapses since a packet arrives to the queue to the time it is "seen" at the receiver. For this purpose, let us first define the mean queue size as $E[Q] = \sum_{i=0}^{B} i\pi_i$ for the case in which the queue capacity is $B$. If there is no constraint on the capacity, we simply let $B \to \infty$. Let us also define $T_{(m,K)}$ as the transmission time for a choice of $(m, K)$. Then,

$$E[T_{(m,K)}] = \left[ t_m \sum_{i=0}^{m} \pi_i + \sum_{i=m+1}^{K-1} t_i \pi_i + t_K \sum_{i=K}^{B} \pi_i \right],$$

where $\sum_{i=K}^{B} \pi_i = 1 - \sum_{i=0}^{K-1} \pi_i$, and if there is no constraint on capacity, again we let $B \to \infty$. Using Little's law,

$$E[D] = E[Q]/\lambda + E[T_{(m,K)}]. \qquad (8)$$

### E. Mean Time Between Decoding Events

One of the key features of an online network coding system is that the data packets can be seen by the receiver without being decoded. We are interested in determining the time between decoding events, which provides us with a metric of the worst case of a packet being seen before it is decoded. We analyze this for the case of $m = 1$ and for a finite capacity queue, but it can be extended for a general $m$ and for the case of an infinite capacity queue.

The time between decoding events can be determined by modelling the problem as an absorbing Markov chain, as in Fig. 2. The absorption state indicates that a decoding event occurred, and we shall considered this to be state 0. Other states correspond to the number of packets in the queue. A transition to the absorbing state may take place from any other state. Let us define $P_a(j|i) = \frac{e^{-\lambda t_i}(\lambda t_i)^j}{j!}$. The transition probability matrix $\mathcal{P}_A$ of this absorbing Markov chain is of the form

$$\mathcal{P}_A = \begin{bmatrix} 1 & 0 & & \cdots & 0 & 0 \\ d_0^{(1)} & d_1^{(1)} & & \cdots & d_{B-1}^{(1)} & Rd(B-1, 1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ d_0^{(K)} & d_1^{(K)} & & \cdots & d_{B-1}^{(K)} & Rd(B-1, K) \\ d_0^{(K)} & 0 & d_1^{(K)} & \cdots & d_{B-2}^{(K)} & Rd(B-2, K) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ d_0^{(K)} & \cdots & d_1^{(K)} & \cdots & d_{B-K}^{(K)} & Rd(B-K, K) \end{bmatrix}$$

where $Rd(k, l) = 1 - \sum_{j=0}^{k} d_j^{(l)}$, and where we define

$$d_j^{(i)} = \sum_{k=\max\{0, j-i\}}^{j-1} P_a(k|i) \mathcal{P}_{(i, \{N_i\}, Pe)}(i + k - j) \qquad (9)$$

for $i \in \{1, ..., K\}$. Finally, $d_0^{(i)} = \mathcal{P}_{(i, \{N_i\}, Pe)}(i)$.

The time between decoding events depends on the starting state. The probability of starting at state $i$ in this absorbing Markov chain depends on the state of the queue and the fact that the previous transmission resulted in packets being seen

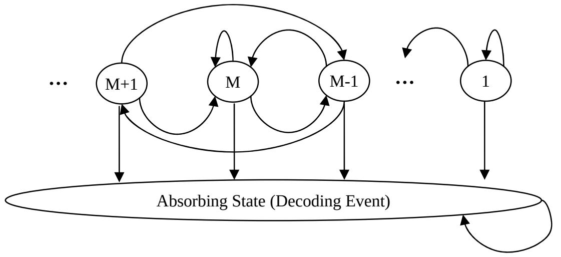

Fig. 2. Absorbing Markov chain to compute time between decoding events.

but not decoded. We determine the probability of starting in state $i$ by using the stationary probabilities of Section 4. Note that every state $i$ in Section 4, inherently contains two states: $(i, d)$ and $(i, n)$, where $d$ and $n$ indicate if the last transmission caused or not a decoding event, respectively. Let us define $\pi_{j,d}$ and $\pi_{j,n}$ as the stationary probabilities associated to the inherent states $(i, d)$ and $(i, n)$, where $\pi_j = \pi_{j,d} + \pi_{j,n}$. Let $P_{i,f \to j,f'}$ be the transition probability from state $(i, f)$ to state $(j, f')$, with $f, f' \in \{d, n\}$. We also define $P_{i \to j,*} = P_{i,d \to j,*}\frac{\pi_{i,d}}{\pi_i} + P_{i,s \to j,*}\frac{\pi_{i,s}}{\pi_i}$ for $* \in \{d, n\}$ as the transition probability from state $i$ (or rather from both $(i, d)$ and $(i, n)$) to state $(j, *)$.

Let us define the time between decoding events as $D_e$. Then, the mean time between decoding events is

$$E[D_e] = \sum_{i \geq 1} E[D_e|i]\pi_{i,n} \tag{10}$$

where $E[D_e|i]$ represents the mean time to absorption in the Markov chain given that the system started in state $i$, where the transition time is $t_i$. Note that

$$\pi_{j,*} = \textstyle\sum_{i,s \in \{d,n\}} P_{i,s \to j,*}\pi_{i,s} = \sum_i P_{i \to j,*}\pi_i \tag{11}$$

for $* \in d, n$. We are interested in determining $\pi_{j,n}$. However, $P_{i \to j,d}$ has a simpler characterization than $P_{i \to j,n}$, given by

$$P_{i \to j,d} = \begin{cases} \mathcal{P}_{(i,\{N_i\},Pe)}(i)P_a(j|i) & \text{if } i \leq K, \\ \mathcal{P}_{(i,\{N_i\},Pe)}(K)P_a(j-(i-K)|K) & \text{if } i > K, \end{cases}$$

and therefore we determine $\pi_{j,d}$ through $P_{i \to j,d}$, and then determine $\pi_{j,n} = \pi_j - \pi_{j,d}$. After some manipulations, we can show that for $j \geq 1$

$$\begin{aligned} \pi_{j,d} = \tfrac{\lambda^j}{j!} \textstyle\sum_{i=1}^{K} \left[\pi_i e^{-\lambda t_i} t_i^j \mathcal{P}_{(i,\{N_i\},Pe)}(i)\right] \\ + \textstyle\sum_{i=K+1}^{B} \tfrac{\pi_i e^{-\lambda t_K}(\lambda t_K)^{j-(i-K)}}{(j-(i-K))!} \left[\mathcal{P}_{(i,\{N_i\},Pe)}(K)\right] \\ \tfrac{e^{-\lambda t_1}(\lambda t_1)^j}{j!}\pi_0 \left[\mathcal{P}_{(1,\{N_i\},Pe)}(1)\right]. \end{aligned} \tag{12}$$

### *F. Performance Analysis*

We study different schemes to illustrate the performance of the system and its relationship to the choice of $N_i$'s.

*1) Optimizing for Mean Delay:* Computes $N_i$'s following an exhaustive search method in order to minimize $E[D]$.

*2) Optimizing for Mean Time between Decoding Events:* Computes $N_i$'s following an exhaustive search method in order to minimize $E[D_e]$.

*3) Heuristics 1:* This heuristics computes $N_i = \lfloor \frac{i}{1-Pe} \rfloor$.

*4) Heuristics 2:* This heuristics computes $N_i = \lceil \frac{i}{1-Pe} \rceil$.

*5) Heuristics 3:* This heuristics computes $N_i$'s so that a minimum criteria for the probability of generating a decoding event in the next transmission is achieved. We specify $\epsilon$ as the minimum acceptable probability of decoding in the next transmission and compute $N_i$'s $\forall i \in m, ..., K$ so that $\mathcal{P}_{(i,\{N_i\},Pe)}(i) \geq \epsilon$.

## V. Numerical Results

This section provides numerical examples that show the performance of our network coding scheme for different values of arrival rate $\lambda$. We use the mean delay and mean time between decoding events as our metrics of interest. We use a high latency channel with packet erasure probability $Pe = 0.1$, propagation time of 12.5 ms, data packets of $10,000$ bits, $g = 20$ bits, a rate $R = 1.5$ Mbps, a header of $h = 80$ bits, $B = 30$ packets, and the ACK packet has 100 bits.

Fig. 3 and 4 show the performance of the different schemes in terms of mean delay and mean time between decoding events, respectively, when the arrival rate is changed. A good mean delay performance does not necessarily imply a poor mean time between decoding events performance, or vice versa. For example, Heuristics 1 for small $\lambda$ shows that it is possible to have poor performance in both metrics. Fig. 3 and 4 also illustrate that the optimal choice of $N_i$'s also depends on arrival rate $\lambda$ for both metrics. However, Heuristics 2 shows a good trade-off between mean delay and mean time between decoding events, while choosing $N_i$'s independently from $\lambda$. For Heuristics 2 the mean time between decoding events is an order of magnitude smaller than its mean delay performance, i.e., the main contribution to the overall delay of a packet, from being received to being decoded, is due to $E[D]$.

Fig. 3 also illustrates that all schemes change smoothly with $\lambda$ in terms of mean delay, except the optimizing for mean time between decoding events scheme which shows a jagged curve. The main reason for this behavior is that the optimal values of $N_i$'s in terms of $E[D_e]$ change considerably with respect to $\lambda$. For small $\lambda$, there might be numerical innacuracies in computing the values of $N_i$'s that minimize $E[D_e]$, due to the very small values reported in Fig. 4. These effects translate into a smooth change in the $E[D_e]$ (Fig. 4) but non-smooth behavior in mean delay.

Finally, note that choosing the $N_i$'s to increase the probability of decoding in the next transmission, as in Heuristics 3, shows poor performance in both metrics and depends on the choice of $\epsilon$. Note that the choice of $\epsilon$ is not directly map into delay performance.

## VI. Conclusions

This paper provides an online network coding scheme for time division duplexing channels with Poisson arrivals, particularly useful for large latency scenarios. The scheme

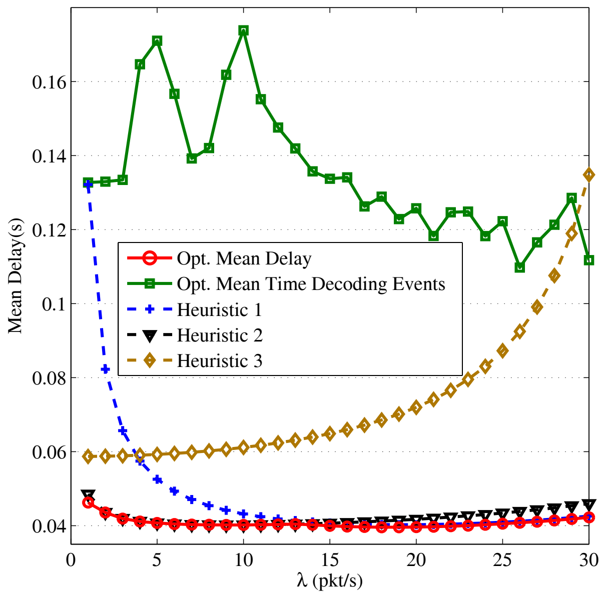

Fig. 3. Mean delay of different schemes for choosing $N_i$'s for the case of $(m, K) = (1, 3)$. For Heuristics 3, $\epsilon = 0.99999$.

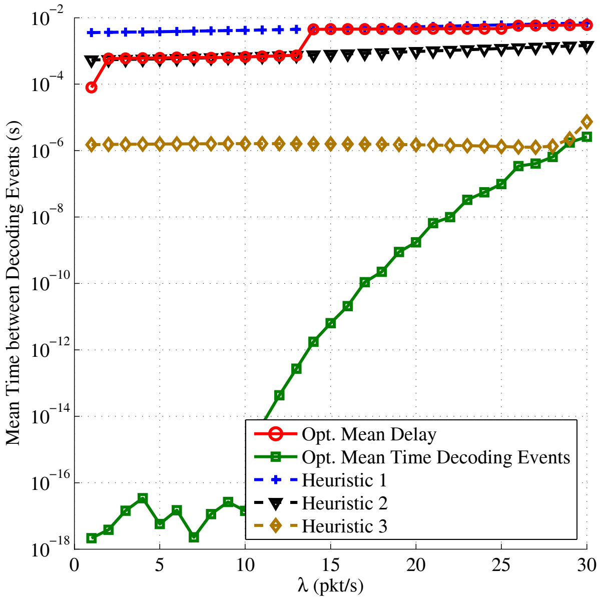

Fig. 4. Mean time between decoding events of different schemes for choosing $N_i$'s for the case of $(m, K) = (1, 3)$.For Heuristics 3, $\epsilon = 0.99999$.

is adaptive in nature, as it relies on a coupling of feedback and coding. A key feature of our system is that the correct amount of redundancy depends not only on system and channel conditions, but also on the need to receive feedback through the same transmission channel. Our analysis considers a queueing model in which the window of packets $M$ that can be combined in any given transmission has a range of values, say $M \in \{m, ..., K\}$. The system is modeled as a bulk queue with feedback with Poisson arrivals and a general service time that depends on the bulk size.

Numerical results suggest that the mean delay and the mean time between decoding events are dependent on the choice of the redundancy, in terms of the number of coded packets that are transmitted back-to-back before stopping to receive an ACK packet, i.e., the $N_i$'s. There is also a dependence on the arrival rate $\lambda$, which is particularly evident if the $N_i$'s are chosen to minimize the mean time between decoding events, affecting both the mean delay performance and the mean time between decoding events.

## Acknowledgment

This work was supported in part by the NSF under grants No. 0520075, 0831728 and CNS-0627021, by ONR MURI Grant # N00014-07-1-0738, and subcontract # 060786 issued by BAE Systems National Security Solutions, Inc. and supported by the Defense Advanced Research Projects Agency (DARPA) and the Space and Naval Warfare System Center (SPAWARSYSCEN), San Diego under Contract # N66001-06-C-2020 (CBMANET).

## References

[1] D. E. Lucani, M. Stojanovic, M. Médard, “Random Linear Network Coding For Time Division Duplexing: When To Stop Talking And Start Listening”, in Proc. INFOCOM 2009, Rio de Janeiro, Brazil, Apr. 2009.

[2] D. E. Lucani, M. Stojanovic, M. Médard, “Random Linear Network Coding For Time Division Duplexing: Energy Analysis”, in Proc. ICC 2009, Dresden, Germany, Jun. 2009.

[3] D. E. Lucani, M. Médard, M. Stojanovic,“On Coding for Delay - New Approaches Based on Network Coding in Networks with Large Latency, in Proc. ITA Workshop, San Diego, Feb. 2009.

[4] D. E. Lucani, M. Médard, M. Stojanovic, “Broadcasting in Time-Division Duplexing: A Random Linear Network Coding Approach”, in Proc. NetCod’09, Lausanne, Switzerland, pp. 62-67, Jun. 2009

[5] D. E. Lucani, M. Médard, M. Stojanovic, D. R. Karger, “Sharing Information in Time-Division Duplexing Channels: A Network Coding Approach”, in Proc. 47th Allerton Conf. on Comm. Cont. and Comp., IL, USA, Sept. 2009.

[6] D. E. Lucani, M. Médard, M. Stojanovic, “Random Linear Network Coding for Time Division Duplexing: Field Size Considerations”, in Proc. GLOBECOM 2009, Hawaii, USA, Dec. 2009.

[7] D. E. Lucani, M. Médard, M. Stojanovic,“Systematic Network Coding for Time Division Duplexing”, submitted to ISIT 2010.

[8] D. E. Lucani, M. Médard, M. Stojanovic,“Random Linear Network Coding for Time Division Duplexing: Queueing Analysis,” in Proc. ISIT 2009, Seoul, Korea, Jun. 2009.

[9] J. Sundararajan, D. Shah, M. Médard, ”On Queueing for Coded Networks - Queue Size Follows Degrees of Freedom”, In Proc. IEEE ITW 2007, Bergen, Norway, Jul. 2007.

[10] B. Shrader, A. Ephremides, “On the Queueing Delay of a Multicast Erasure Channel”, In Proc. IEEE ITW 2006, Oct. 2006.

[11] B. Shrader, A. Ephremides,“A Queueing Model for Random Linear Coding”, In Proc. IEEE MILCOM, Oct. 2007.

[12] M. L. Chaudhry, J. G. C. Templeton,“A First Course in Bulk Queues”, Wiley, 1983.

[13] S. K. Bar-Lev, M. Parlar, D. Perry, W. Stadje, F. A. Van der Duyn Schouten, “Applications of Bulk Queues to Group Testing Models with Incomplete Identification”, Euro. Jour. of Oper. Res., no. 183, pp. 226237, 2007.

[14] L. E. N. Delbrouck, “A Feedback Queueing System With Batch Arrivals, Bulk Service, and Queue-Dependent Service Time,” Jour. of the Assoc. for Comp. Mach., vol. 17, no. 2, pp. 314-323, 1970.

[15] T. Ho, M. Médard, R. Koetter, D.R. Karger, M. Effros, J. Shi, B. Leong,“A Random Linear Network Coding Approach to Multicast”, Trans. Info. Theory, vol. 52, no. 10, pp.4413-4430, Oct. 2006.

[16] W.B. Powell, “Stochastic Delays in Transportation Terminals: New Results in the Theory and Application of Bulk Queues,” PhD thesis, MIT, 1981.