

On Coding for Delay - New Approaches Based on Network Coding in Networks with Large Latency

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—In networks with large latency, feedback about received packets may lag considerably the transmission of the original packets, limiting the feedback’s usefulness. Moreover, time duplex constraints may entail that receiving feedback may be costly. In this work, we consider tailoring feedback and coding jointly in such settings to reduce the expected delay for successful in order reception of packets. We find that, in certain applications, judicious choices provide results that are close to those that would be obtained with a full-duplex system.

I. INTRODUCTION

The concept of network coding, also known as coded packet networks, was introduced by Ahlswede *et al* [1]. Network coding considers nodes that have a set of functions that operate upon received or generated data packets. A classical network’s task is to transport packets provided by the source nodes unmodified, i.e. they constitute a subset of the coded packet networks, in which each node has two main functions: forwarding and replicating a packet. In contrast, network coding considers information as an algebraic entity, on which one can operate.

Reference [2] considered, for the first time, the use of network coding in channels in which time division duplexing is necessary, i.e. when a node can only transmit or receive, but not both at the same time. This type of channel is usually called half-duplex, but we will use the more general term time division duplexing (TDD) to emphasize the fact that the transmitter and receiver do not use the channel in any pre-determined fashion, but instead may *vary* the amount of time allocated to transmit and receive. Important examples of time division duplexing channels are infrared devices (IrDA), which have motivated many TDD ARQ schemes [3], and underwater acoustic communications [4]. Other important applications may be found in channels with very high latency, e.g. in satellite [5], and deep space [6] communications.

In particular, Reference [2] studied the problem of transmitting M data packets through a link using random linear network coding with the objective of minimizing the expected time to complete transmission of the M data packets. Reference [7] focused on the problem of energy consumption of the scheme showing that there exists, under the minimum energy criterion, an optimal number of coded data packets to be transmitted back-to-back before stopping to wait for an acknowledgment (ACK).

We present a full characterization of the problem for both time and energy to complete transmission for this scheme by means of the moment generating function of these random variables. Using this moment generating function we provide expressions for the mean and the variance. We provide a numerical method to compute the first negative moment of the completion time, which is useful to determine the mean throughput of our scheme.

We present an analysis and numerical results that show that transmitting the optimal number of coded data packets sent before stopping to listen for an ACK as in [2] provides performance very close to that of a network coding scheme operating in a full duplex channel, in terms of mean time to complete transmission of all packets. This is the case even in high latency channels. Choosing a number different from the optimum can cause a large degradation in performance, especially if latency is high.

We also show that choosing the number of coded data packets to optimize mean completion time, as in [2], consumes much less energy on average than a full duplex network coding scheme, and only slightly more energy than when we choose the number of coded packets to minimize completion energy as in [7]. Thus, our scheme provides a good trade-off between energy consumption and completion time.

The paper is organized as follows. In Section 2, we outline the problem. In Section 3, we derive the moment generating function for the completion time and energy. In Section 4, we present the analysis of mean time and mean energy to complete transmission of M data packets and the optimization required to determine the number of coded packets to transmit before stopping in order to minimize mean completion time or mean completion energy. Section 5 studies the expressions of the variance for both the completion time and energy. In Section 6, the throughput performance is analyzed. Section 7 presents the schemes to be used for performance evaluation, while Section 8 provides numerical results. Conclusions are summarized in Section 9.

II. RANDOM NETWORK CODING FOR TDD CHANNELS

This section provides a review of network coding for TDD channels presented in [2]. We consider a sender in a link that wants to transmit M data packets at a given link data rate R . The channel is modeled as a packet erasure channel.

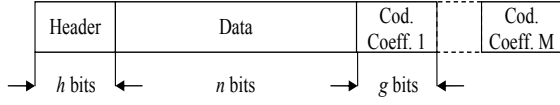


Fig. 1. [2] Structure of coded data packet.

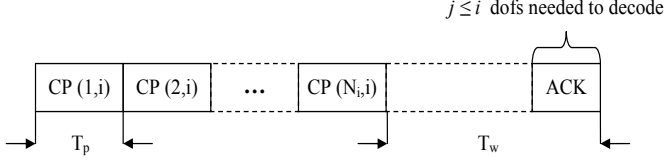


Fig. 2. [2] Network coding TDD scheme.

Nodes can only transmit or receive, but not both at the same time. The sender uses random linear network coding [8] to generate coded data packets. Each coded data packet contains a linear combination of the M data packets of n bits each, as well as the random encoding coefficients used in the linear combination. Each coefficient is represented by g bits. For encoding over a field size q , we have that $g = \log_2 q$ bits. Also consider an information header of size h . Thus, the total number of bits per packet is $h + n + gM$. Figure 1 shows the structure of each coded packet consider in our scheme.

The sender can transmit coded packets back-to-back before stopping to wait for the ACK packet. The ACK packet feeds back the number of degrees of freedom (dof), that are still required to decode successfully the M data packets. Since random linear coding is used, there is some probability of choosing encoding vectors that are all zero for one coded packet or encoding vectors that are linearly dependent on vectors of previously received packets. Thus, the expected number of successfully received packets before having M linearly independent combinations is [2]

$$\sum_{k=1}^M \frac{1}{(1 - (1/q)^k)} \leq M \frac{q}{q-1}. \quad (1)$$

In the following analysis, we assume that the field size q is large enough so that the expected number of successfully received packets at the receiver, in order to decode the original data packets, is approximately M . This is not a necessary assumption for our analysis. We could have included the probabilities of receiving linearly independent combinations into the transition probabilities. However, making this assumption simplifies the expressions and provides a good approximation for large enough q .

We are interested in determining the optimal number of coded packets that should be sent back-to-back before waiting for an ACK packet from the receiver in order to minimize the time for successfully transmitting the M data packets over the link.

Note that if M packets are in the queue, at least M degrees of freedom have to be sent in the initial transmission, i.e.

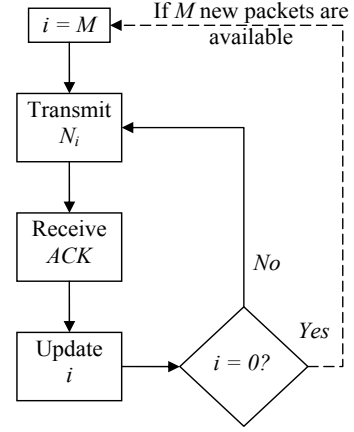


Fig. 3. [2] Network coding algorithm for time division duplexing channels. i represents the remaining number of degrees of freedom to decode the packets, and N_i the corresponding number of coded packets transmitted before stopping to listen for a new ACK. The ACK packet has the information to update i .

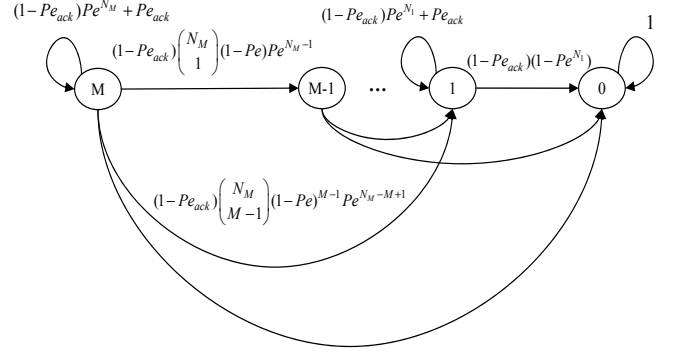


Fig. 4. [2] Markov chain representation of the scheme. State i represents that the receiver requires i more successfully received coded packets to decode the information.

$N_M \geq M$ coded packets. We are interested not only in the number of dof that are required at the first transmission, but also at subsequent stages. Transmission begins with M information packets, which are encoded into N_M random linear coded packets and transmitted. If all M packets are decoded successfully, the process is completed. Otherwise, the ACK informs the transmitter how many are missing, say i . The transmitter then sends N_i coded packets, and so on, until all M packets have been decoded successfully. We are interested in the optimal number N_i of coded packets to be transmitted back-to-back in the next transmission to complete the remaining i dofs. Figure 3 shows the communication process as a system transmits N_M coded packets initially and awaits reception of an ACK packet that updates the value of i , at which point it will transmit N_i coded packets. The system will keep transmitting and stopping to update i , until $i = 0$. When $i = 0$, the transmitter can start with M new data packets or simply stop. In Figure 2, $CP(k, d)$ represents the k -th coded packet transmitted when we start transmission with

d dofs needed at the receiver to decode the information.

The process can be modelled as a Markov Chain (Figure 4). The states are defined as the number of dofs required at the receiver to decode successfully the M packets. Thus, these states range from M to 0. This is a Markov Chain with M transient states and one recurrent state (state 0). Let us define N_i as the number of coded packets that are sent when i dofs are required at the receiver in order to decode the information. Note that the time spent in each state depends on the state itself, because $N_i \neq N_j, \forall i \neq j$ in general.

The transition probabilities from state i to state j ($P_{i \rightarrow j}$) have the following expression for $0 < j < i$ and $N_i \geq i$:

$$P_{i \rightarrow j} = (1 - Pe_{ack}) \binom{N_i}{i-j} (1 - Pe)^{i-j} Pe^{N_i-i+j} \quad (2)$$

where Pe and Pe_{ack} represents the erasure probability of a coded packet and of an ACK packet, respectively.

More generally, the transition probability can be defined for any value of $N_i \geq 1$ as follows:

$$P_{i \rightarrow j} = (1 - Pe_{ack}) f(i, j) (1 - Pe)^{i-j} Pe^{N_i-i+j} \quad (3)$$

where

$$f(i, j) = \begin{cases} \binom{N_i}{i-j} & \text{if } N_i \geq i, \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

For $j = i$ the expression for the transition probability reduces to:

$$P_{i \rightarrow i} = (1 - Pe_{ack}) Pe^{N_i} + Pe_{ack} \quad (5)$$

For completeness, note that $P_{i \rightarrow 0} = 1 - \sum_{j=1}^i P_{i \rightarrow j}, \forall i$ and $P_{0 \rightarrow 0} = 1$.

III. MOMENT GENERATING FUNCTION

Let us define the moment generating function of the completion time when the Markov Chain starts at state n as

$$M_{T,n}(s) = \sum_t \exp(st) P_T(T = t) \quad (6)$$

where $P_T(T = t)$ is the probability of the completion time being t . Note that $M_{T,n}(s)$ is the moment generating function of the completion time when n data packets are taken by the source to be transmitted reliably to the receiver.

Using the Markov Chain structure of the problem, it can be shown that $M_{T,n}(s)$ can be re-stated as

$$M_{T,n}(s) = \sum_{m_n \geq 1} \sum_{m_{n-1} \geq 0} \cdots \sum_{m_1 \geq 0} \exp \left(s \sum_{i=1}^n m_i T^i \right) C_n A_n \quad (7)$$

where T^i is the deterministic time required to send N_i coded packets and wait for an ACK when the Markov chain is in state i , i.e. $T^i = N_i T_p + T_w$, where T_p is the transmission time of a coded packet, and T_w is the waiting time to receive an

ACK packet, as shown in Figure 2. The constant C_n captures the effect of returning to the same state repeatedly, while A_n captures the different paths that can be traversed without repetition of a state.

The expression for C_n is

$$C_n = \prod_{j=1}^n P_{j \rightarrow j}^{m_j-1}.$$

The coefficient for A_n can be shown to obey a recursive expression of the form

$$A_n = \mathbf{1}_{\{m_n > 0\}} \left[\sum_{j=0}^{n-1} P_{n \rightarrow j} \left(\prod_{i=j+1}^{n-1} P_{i \rightarrow i} \mathbf{1}_{\{m_i=0\}} \right) A_j \right]$$

with $A_1 = P_{1 \rightarrow 0} \mathbf{1}_{\{m_1 > 0\}}$. The indicator function $\mathbf{1}_{\{s \in S\}}$ is 1 when $s \in S$ and zero otherwise.

Substituting expression (8) into (7) we obtain the following recursive equation for the moment generating function

$$M_{T,n}(s) = \frac{\exp(sT^n)}{1 - P_{n \rightarrow n} \exp(sT^n)} \sum_{i=0}^{n-1} P_{n \rightarrow i} M_{T,i}(s) \quad (8)$$

with $M_{T,0}(s) = 1$.

Finally, note that the same structure is valid for computing the energy needed to complete transmission. To do so, one would substitute T^i by E^i , and $M_{T,n}(s)$ by $M_{E,n}(s)$, which leads to

$$M_{E,n}(s) = \frac{\exp(sE^n)}{1 - P_{n \rightarrow n} \exp(sE^n)} \sum_{i=0}^{n-1} P_{n \rightarrow i} M_{E,i}(s) \quad (9)$$

with $M_{E,0}(s) = 1$.

IV. MEAN COMPLETION TIME AND ENERGY

The expected time for completing the transmission of the M data packets constitutes the expected time of absorption, i.e. the time to reach state 0 for the first time, given that the initial state is M . This can be expressed in terms of the expected time for completing the transmission given that the Markov Chain is in state i , T_i , $\forall i = 0, 1, \dots, M-1$.

By taking the first derivative of the moment generating function, it can be easily proven that

$$T_i = \frac{\partial M_{T,n}(s)}{\partial s} \Big|_{s=0} = \frac{T^n + \sum_{i=1}^{n-1} P_{n \rightarrow i} \frac{\partial M_{T,i}(s)}{\partial s} \Big|_{s=0}}{1 - P_{n \rightarrow n}} \quad (10)$$

where $T^i = N_i T_p + T_w$ as in Section III.

For our scheme, $T_p = \frac{h+n+gM}{R}$ and $T_w = T_{rt} + T_{ack}$, where $T_{ack} = n_{ack}/R$, n_{ack} is the number of bits in the ACK packet, R is the link data rate, and T_{rt} is the round trip time. Note that $T_0 = 0$. Then, for $i > 1$:

$$T_i = \frac{N_i T_p + T_w}{(1 - Pe_{ack})(1 - Pe^{N_i})} + \frac{(1 - Pe)^i Pe^{N_i-i} \sum_{j=1}^{i-1} f(i, j) \left(\frac{Pe}{1 - Pe} \right)^j T_j}{1 - Pe^{N_i}}. \quad (11)$$

For example, for $i = 1$ we have that:

$$T_1 = \frac{(N_1 T_p + T_w)}{(1 - P_{e_{ack}})(1 - P e^{N_1})}. \quad (12)$$

As it can be seen, the expected time for each state i depends on all the expected times for the previous states. Because of the Markov property, we can optimize the values of all N_i 's in a recursive fashion, i.e. starting by N_1 , then N_2 and so on, until N_M , in order to minimize the expected transmission time. We do so in the following subsection.

Using a similar argument, we show that the mean completion energy $E_i, \forall i = 1, \dots, M$ is

$$E_i = \frac{E^i}{(1 - P_{e_{ack}})(1 - P e^{N_i})} + \frac{(1 - P e)^i P e^{N_i - i} \sum_{j=1}^{i-1} f(i, j) \left(\frac{P e}{1 - P e} \right)^j E_j}{1 - P e^{N_i}} \quad (13)$$

where E^i is the energy consumed by the system to transmit N_i packets and receive an ACK.

For this analysis, we consider the case of $E^i = N_i E_p + E_{ack}$, where E_p is the transmission energy of a coded packet, and E_{ack} is the transmission energy of an ACK packet. That is, we consider the case in which transmission energy is dominant in the total energy consumption E^i . In other words, the energy used at the receiver and transmitter while waiting for a coded packet and a ACK, respectively, is negligible.

More specifically, we define $E_p = P T_p$, P is the transmission power, and $E_{ack} = P T_{ack}$.

A. Minimizing Mean Completion Time

Our objective is to minimize the value of the expected transmission time T_M . Without assuming any particular value for N_i , we have that

$$\begin{aligned} \min_{N_M, \dots, N_1} T_M &= \\ &= \min_{N_M} \frac{N_M T_p + T_w}{(1 - P_{e_{ack}})(1 - P e^{N_M})} \\ &+ \frac{(1 - P e)^M P e^{N_M - M} \sum_{j=1}^{M-1} f(M, j) \left(\frac{P e}{1 - P e} \right)^j \min_{N_j, \dots, N_1} T_j}{1 - P e^{N_M}} \end{aligned} \quad (14)$$

Hence, regardless of the assumption on N_i , the problem of minimizing T_M in terms of the variables N_M, \dots, N_1 can be solved iteratively. First, we compute $\min_{N_1} T_1$, then use this results in the computation of $\min_{N_2, N_1} T_2$, and so on.

One approach to computing the optimal values of N_i is to ignore the constraint to integer values and take the derivative of T_i with respect to N_i and look for the value that sets it equal to zero. For our particular problem, this approach leads to solutions without a closed form, i.e. expressed as an implicit function. For $M = 1$, the optimal value of N_1 can be expressed using a known implicit function (Lambert function), and it is given by

$$N_1^* = \frac{1 + W \left(-\exp \left(-1 + \frac{\ln(P e) T_w}{T_p} \right) \right)}{\ln P e} - \frac{T_w}{T_p} \quad (15)$$

where $W(\cdot)$ is the Lambert W function [11]. The positive values are found for the branch W_{-1} , as denoted in reference [11].

The case of $M = 1$ can be thought of as an optimized version of the uncoded Stop-and-Wait ARQ, which is similar to the idea presented in [5]. Instead of transmitting one packet and waiting for the ACK, our analysis suggests that there is an optimal number of back-to-back repetitions of the same data packet that should be transmitted before stopping to listen for an ACK packet.

Instead of using the previous approach, we perform a search for the optimal values $N_i, \forall i \in \{1, \dots, M\}$, using integer values. Thus, the optimal N_i 's can be computed numerically for given $P e$, $P_{e_{ack}}$, T_w and T_p . In particular, the search method for the optimal value can be made much simpler by exploiting the recursive characteristic of the problem, i.e. instead of making a M -dimensional search, we can perform M one-dimensional searches. Finally, these N_i 's do not need to be computed in real time. They can be pre-computed for different channel conditions (e.g. $P e$, T_{rt}) or system settings (e.g. n , M , g , data rate), and stored in the receiver as look-up tables. This procedure makes the computational load on the nodes to be negligible at the time of determining the optimal number of coded packets in terms of the completion time, especially for dynamic environments.

B. Minimizing Mean Completion Energy

In this case, our objective is to minimize the value of the mean completion energy E_M , that is

$$\begin{aligned} \min_{N_M, \dots, N_1} E_M &= \min_{N_M} \frac{N_M E_p + E_{ack}}{(1 - P_{e_{ack}})(1 - P e^{N_M})} \\ &+ \frac{(1 - P e)^M P e^{N_M - M} \sum_{j=1}^{M-1} f(M, j) \left(\frac{P e}{1 - P e} \right)^j \min_{N_j, \dots, N_1} E_j}{1 - P e^{N_M}} \end{aligned}$$

which is very similar to the result of T_M making the appropriate substitutions.

The search method proposed to determine the N_i values in order to minimize T_M is valid for E_M . Reference [7] studies this problem in more detail.

V. VARIANCE

Another figure of importance is the variance of the completion time and energy. We can use the moment generating function for our problem knowing that

$$Var_{T,n} = \frac{\partial^2 M_{T,n}(s)}{\partial s^2} \Big|_{s=0} - \left(\frac{\partial M_{T,n}(s)}{\partial s} \Big|_{s=0} \right)^2 \quad (16)$$

where $Var_{T,n}$ is the variance of T when $M = n$.

By taking derivatives, it is possible to prove that

$$\begin{aligned} \frac{\partial^2 M_{T,n}(s)}{\partial s^2} \Big|_{s=0} &= \frac{2 T^n}{1 - P_{n \rightarrow n}} \frac{\partial M_{T,n}(s)}{\partial s} \Big|_{s=0} - \frac{(T^n)^2}{1 - P_{n \rightarrow n}} \\ &+ \frac{1}{1 - P_{n \rightarrow n}} \sum_{i=1}^{n-1} P_{n \rightarrow i} \frac{\partial^2 M_{T,i}(s)}{\partial s^2} \Big|_{s=0} \end{aligned} \quad (17)$$

Again, we can substitute the values of $T^i, \forall i$, and the values of the transition probabilities in order to compute the variance.

Note that the same results apply for the case of energy making the appropriate substitutions of T^i by E^i , and $M_{T,i}$ by $M_{E,i}$.

VI. THROUGHPUT

The mean throughput for our block scheme can be defined as

$$\text{Mean Throughput} = E\left[\frac{Mn}{T}\right] = MnE\left[\frac{1}{T}\right] \quad (18)$$

where we assume M and n to be constants.

This implies that the problem of computing the mean throughput for our scheme is equivalent to that of computing negative moments of the completion time. The problem of computing negative integer moments has been studied previously in [9] and [10]. In particular, we focus in the result of [10] which states that

$$E[X^{-1}] = \int_0^\infty M_X(-s)ds \quad (19)$$

where $X > 0$ is the random variable, and $M_X(s)$ is the moment generating function of X .

Note that for the case of $M = 1$ we can compute $E[T^{-1}]$ by direct computation of this random variable or by using expression (19). Using direct computation

$$E[T^{-1}] = \frac{P_{1 \rightarrow 0}}{P_{1 \rightarrow 1}} \sum_{k=1}^\infty \frac{P_{1 \rightarrow 1}^k}{kT^1} \quad (20)$$

$$= \frac{P_{1 \rightarrow 0}}{P_{1 \rightarrow 1}T^1} \sum_{k=1}^\infty \frac{(1-P_{1 \rightarrow 0})^k}{k} \quad (21)$$

$$= \frac{P_{1 \rightarrow 0}}{P_{1 \rightarrow 1}T^1} \ln\left(\frac{1}{P_{1 \rightarrow 0}}\right) \quad (22)$$

where we have used the Mercator series since $|1 - P_{1 \rightarrow 0}| < 1$ for all cases of interest [2]. If we use expression (19) we obtain

$$E[T^{-1}] = \int_0^\infty M_{T,1}(-s)ds \quad (23)$$

$$= \int_0^\infty P_{1 \rightarrow 0} \frac{\exp(sT^1)}{1 - P_{1 \rightarrow 1} \exp(sT^1)} ds \quad (24)$$

$$= \frac{P_{1 \rightarrow 0}}{T^1 P_{1 \rightarrow 1}} \int_{1-P_{1 \rightarrow 1}}^1 \frac{du}{u} = \frac{P_{1 \rightarrow 0}}{P_{1 \rightarrow 1}T^1} \ln\left(\frac{1}{P_{1 \rightarrow 0}}\right) \quad (25)$$

where we have used the fact that $P_{1 \rightarrow 0} = 1 - P_{1 \rightarrow 1}$. In both cases we get the same result.

For $M > 1$, these expressions are complicated using direct computation. However, it is possible to compute them if we use expression (19) and the structure of the moment generating function of our problem (Expression (8)). For the case of $M = j$ we get

$$E[T^{-1}] = \int_0^\infty \frac{\exp(-sT^j)}{1 - P_{j \rightarrow j} \exp(-sT^j)} \sum_{i=0}^{j-1} P_{j \rightarrow i} M_{T,i}(-s) ds. \quad (26)$$

Notice that $M_{T,i}(-s), \forall i$ have a multiplying term $\frac{\exp(-sT^i)}{1 - P_{i \rightarrow i} \exp(-sT^i)}$ which decreases to zero exponentially as $s \rightarrow \infty$ and goes to $\frac{1}{1 - P_{i \rightarrow i}}$ as $s \rightarrow 0$. Thus, all terms inside the integral in (27) will go to zero exponentially.

Using this characteristic we can numerically compute $E[T^{-1}]$ using numerical integration techniques with the following approximation

$$E[T^{-1}] \approx \int_0^\tau \frac{\exp(-sT^j)}{1 - P_{j \rightarrow j} \exp(-sT^j)} \sum_{i=0}^{j-1} P_{j \rightarrow i} M_{T,i}(-s) ds. \quad (27)$$

where $\tau = \max_{i=1, \dots, j} \tau_i$, $\tau_i = C/T^i$, and C is a constant in order to ensure $\exp(-\tau_i T^i)$ is small enough, e.g. $C = 5$ ensures $\exp(-\tau_i T^i) = \exp(-5) \approx 0.0067$.

Although this measure is important, we will define a different throughput measure called η because 1) the mean throughput is computationally demanding, and 2) most of the analysis of typical ARQ schemes is performed using η .

Let us define our measure of throughput η as the ratio between number of data bits transmitted (n) and the time it takes to transmit them. For the case of a block-by-block transmission, as described in Section II,

$$\eta = \frac{Mn}{T_M} \quad (28)$$

where T_M is the expected time of completion defined previously.

Note that the mean throughput and η are not equal. For the case of $M = 1$, note that $E[\frac{Mn}{T}] = \eta \frac{\ln(1/P_{1 \rightarrow 0})}{P_{1 \rightarrow 1}}$. More generally, using Jensen's inequality, $MnE[\frac{1}{T}] \geq \frac{Mn}{T_M}$ for $T > 0$. Therefore, η constitutes a lower bound to the mean throughput in our scheme.

Also, note that if M and n are fixed, η is maximized as T_M is minimized. Thus, by minimizing the mean time to complete transmission of a block of M data packets with n bits each, we are also maximizing η for those values. However, we show that the maximal η should be obtained using M and n as arguments in our optimization.

This is important for systems in which the data is streamed. In this case, searching for the optimal values of M and n , in terms of η , provides a way to optimally divide data into blocks of M packets with n bits each before starting communication using our scheme.

A. Optimal Packet Size and Number of Packets per Block

We have discussed throughput with a pre-determined choice of the number of data bits n and the number of data packets M in each block. However, expression 28 implies that the throughput η depends on both n and M . Hence, it is possible to choose these parameters so as to maximize the throughput [2]. We can approach this problem in several ways. The first approach is to look for the optimal n while keeping M fixed:

$$\eta_{opt}(M) = \arg \max_n \left\{ \max_{N_M, \dots, N_1} \eta \right\} \quad (29)$$

The second approach is to look for the optimal M while keeping n fixed:

$$\eta_{opt}(n) = \arg \max_M \left\{ \max_{N_M, \dots, N_1} \eta \right\} \quad (30)$$

More generally, we could consider the case in which both parameters are variable and we are interested in maximizing η :

$$\eta_{opt} = \arg \max_{n, M} \left\{ \max_{N_M, \dots, N_1} \eta \right\} \quad (31)$$

VII. PERFORMANCE EVALUATION

For this study, five schemes are considered. The first two schemes correspond to two network coding TDD schemes that optimize mean time to complete transmission (TDD-T) and mean energy consumption (TDD-E). The third is a full duplex scheme presented in [2] and [7]. The final two schemes are typical TDD ARQ schemes: Go-back-N (GBN) and Selective Repeat (SR). Let us explain in more detail each of the schemes.

1) Network coding for TDD optimized for mean completion time (TDD-T): This is our TDD scheme when we choose the N_i s to optimize the mean completion time given channel characteristics and system parameters.

2) Network coding for TDD optimized for mean completion energy (TDD-E): This is our TDD scheme when we choose the N_i s to optimize the mean completion energy given channel characteristics and system parameters.

3) Network coding in full duplex: This scheme assumes that nodes are capable of receiving and transmitting information simultaneously, and in that sense it is optimal in light of minimal delay. The sender transmits coded packets back-to-back until an ACK packet for correct decoding of all information (M information packets) has been received. This scheme can be modeled as a Markov chain where, as before, the states represent the number of dofs received. The time spent in each state is the same (T_p). Once the M packets have been decoded, i.e. M dofs have been received, the receiver transmits ACK packets back-to-back, each of duration T_{ack} . One ACK should suffice but this procedure minimizes the effect of a lost ACK packet.

The mean time to complete the transmission and get and ACK is [2]:

$$E[T] = T_{rt} + \frac{MT_p}{1 - P_e} + \frac{T_{ack}}{1 - P_{e_{ack}}} \quad (32)$$

where T is the time to complete transmission of M packets.

The mean energy to complete the transmission and get and ACK is [7]:

$$E[\text{Energy}] = \frac{T_{rt}E_p}{T_p} + \frac{T_{rt}E_{ack}}{2T_{ack}} + \frac{ME_p}{1 - P_e} + \frac{E_{ack}}{1 - P_{e_{ack}}} \quad (33)$$

4) Go-Back-N ARQ for TDD: This is an ARQ scheme developed for a TDD duplex channel studied extensively in [3]. Each transmission contains W data packets sent back-to-back, where W is the window size of our GBN scheme. Reference [3] studied this case and proposed the utilization factor for it. In our notation, the equivalent η is given by

$$\eta_{GBN} = \frac{n(1 - P_e) \left(1 - (1 - P_e)^W \right)}{(WT_p + T_w)P_e}. \quad (34)$$

5) Selective repeat ARQ for TDD: This is an ARQ scheme developed for a TDD duplex channel presented in [3]. Each

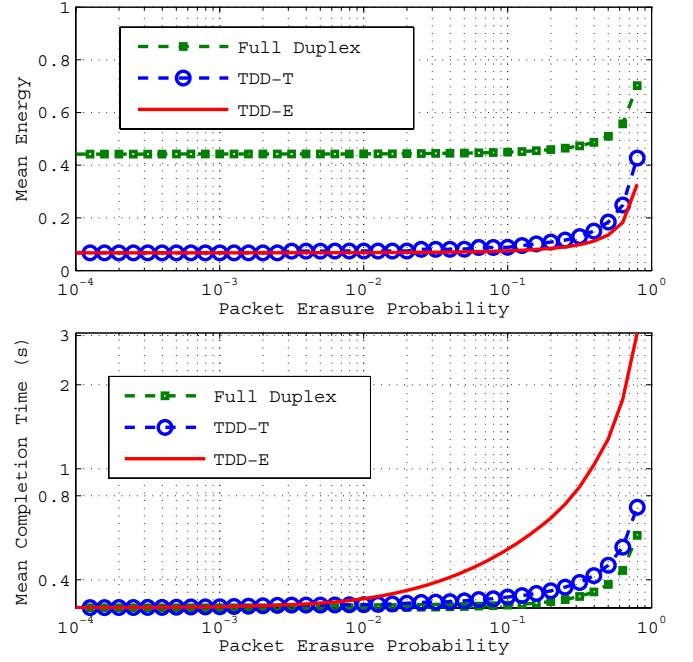


Fig. 5. [7] Mean Energy and Time to complete transmission. Parameters used: $M = 10$, packet size $n = 10,000$ bits, $R = 1.5$ Mbps, $h = 80$ bits, $g = 20$ bits, $n_{ack} = 100$ bits.

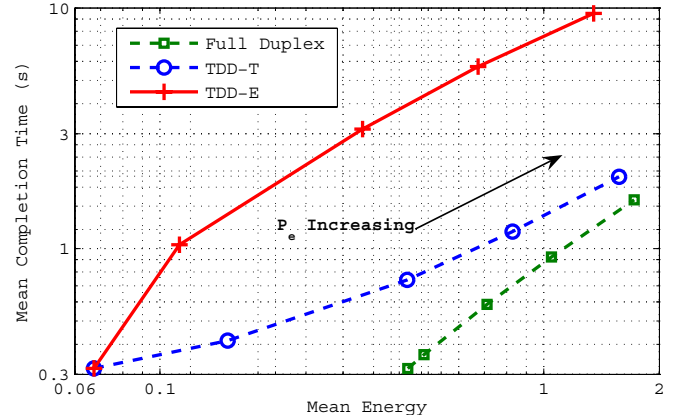


Fig. 6. [7] Mean Energy and Time to complete transmission trade-off. Parameters used: $M = 10$, packet size $n = 10,000$ bits, $R = 1.5$ Mbps, $h = 80$ bits, $g = 20$ bits, $n_{ack} = 100$ bits, and $P_e = 0.00001, 0.4, 0.8, 0.9, 0.95$.

transmission contains W data packets, where W is the window size of our SR scheme. Using the utilization factor studied in Reference [3], we provided the equivalent η in our notation [2]

$$\eta_{SR} = \frac{Wn(1 - P_e)}{WT_p + T_w}. \quad (35)$$

VIII. NUMERICAL RESULTS

This section provides numerical examples that compare the performance of the different network coding schemes we have discussed so far, namely the two TDD schemes that

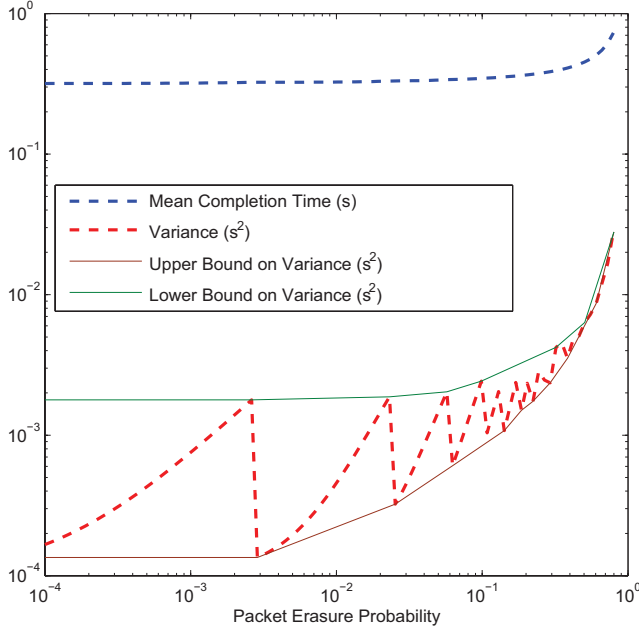


Fig. 7. Variance and Mean of completion time for TDD-T versus packet erasure probability Pe , with parameters $g = 20$ bits, $n_{ack} = 100$ bits, $h = 80$ bits, data rate 1.5 Mbps, $T_{rt} = 250$ ms, $Pe_{ack} = 0.001$, $M = 10$, and $n = 10,000$ bits.

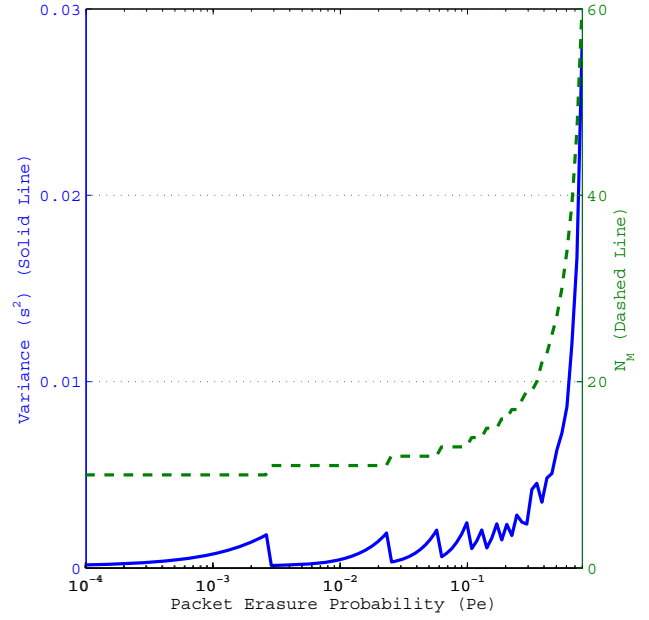


Fig. 8. Variance of completion time and N_M for TDD-T versus packet erasure probability Pe , with parameters $g = 20$ bits, $n_{ack} = 100$ bits, $h = 80$ bits, data rate 1.5 Mbps, $T_{rt} = 250$ ms, $Pe_{ack} = 0.001$, $M = 10$, and $n = 10,000$ bits.

optimize mean energy consumption (TDD-E) and mean time to complete transmission (TDD-T), and a full duplex scheme. The comparison is carried out in terms of the mean energy and mean time to complete transmission of M data packets under different packet erasure probabilities, with the objective of showing the trade-off between energy and completion time of the different schemes. We also present results in terms of the measure of throughput η to illustrate its dependence on the values of M and n for varying channel characteristics (erasure probabilities). We use the case of satellite communications as an example of high latency channels.

Figure 5 studies the mean energy and time to complete transmission of $M = 10$ data packets of size $n = 10,000$ bits, with different packet erasure probabilities in a GEO satellite link with a propagation delay of 125 ms, i.e. $T_{rt} = 250$ ms. In the following results, we have considered that coded packets and ACK are transmitted with the same power, and that this value is normalized, i.e. $P = 1$. The link parameters are specified in the Figure.

The first thing to notice in Figure 5 is that both TDD schemes have much better performance with respect to the full duplex scheme, i.e. energy consumption of the full duplex scheme is considerably higher than the TDD schemes given the high latency characteristic of this channel.

Figure 5 shows that the gap between our network coding scheme optimized for energy and for completion time. Their performance stays similar over a wide range of packet erasure probabilities. When the packet erasure probability is low, the performance is the same for the two approaches, both in the sense of energy and delay. For high packet erasure probability

the performance of both TDD versions is similar in terms of energy, although we observe a clear advantage of TDD-T over TDD-E in mean completion time.

Figure 5 also illustrates that our network coding scheme optimized for completion time (TDD-T) and the network coding full duplex optimal scheme have similar performance over a wide range of packet erasure probabilities. In fact, for the worst case ($Pe = 0.8$) presented in this Figure, our scheme has an expected time of completion only 30 % above the full duplex scheme. Thus, TDD-T can have similar performance to that of full duplex optimal scheme, in the sense of expected time to completion, while showing similar performance to TDD-E, the version optimized for energy consumption. This means that the TDD-T provides a good trade-off between energy and time to complete transmissions.

Let us study the variance of the TDD-T scheme under different erasure probabilities. Figure 7 shows that the variance is very small but it is not a continuous function, showing discontinuities for certain values of Pe . Figure 8 shows that this discontinuities are related to a change in the number of coded packets sent in the first transmission of each M blocks, i.e. N_M . The variance decreases when N_M increases because we are increasing the probability of decoding all M packets after the first transmission. In practice, the Pe is an estimate of the packet erasure probability and these discontinuities can be misleading in terms of expected system performance. Thus, having bounds on the variance for each Pe , as shown in Figure 7, is more meaningful from a system's perspective.

Let us compare the mean throughput $MnE[1/T]$ and $\eta = Mn/E[T]$. Figure 9 shows that both $E[1/T]$ and $1/E[T]$ are

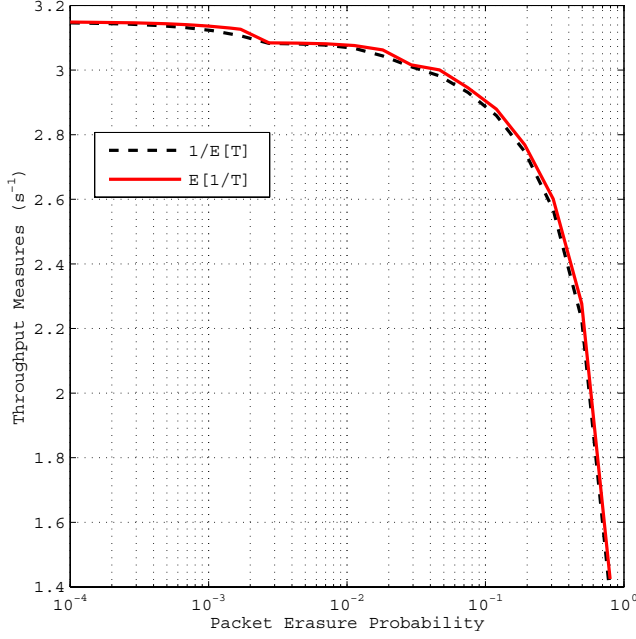


Fig. 9. Mean throughput and η versus Pe for TDD-T with parameters $g = 20$ bits, $n_{ack} = 100$ bits, $h = 80$ bits, data rate 1.5 Mbps, $T_{rt} = 250$ ms, $Pe_{bit} = 0.0001$, $M = 10$, and $n = 10,000$ bits.

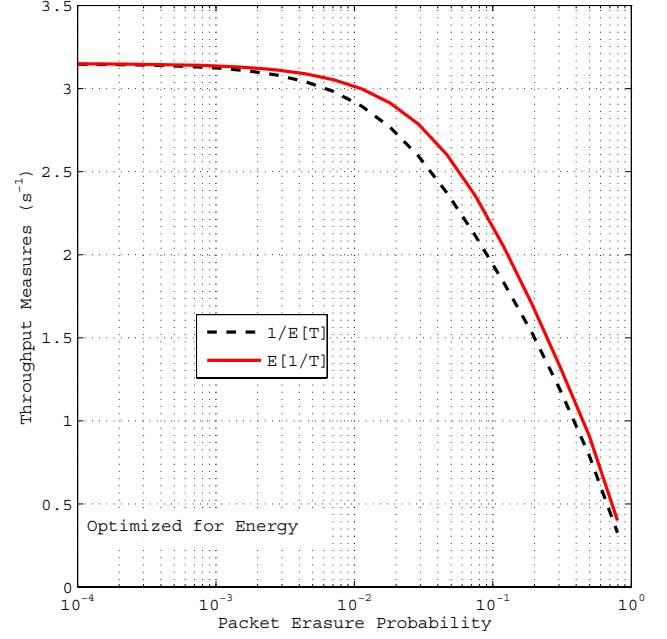


Fig. 10. Mean throughput and η versus Pe for TDD-E with parameters $g = 20$ bits, $n_{ack} = 100$ bits, $h = 80$ bits, data rate 1.5 Mbps, $T_{rt} = 250$ ms, $Pe_{bit} = 0.0001$, $M = 10$, and $n = 10,000$ bits.

very close when we optimize the N_i s in terms of the mean completion time. Thus, choosing the parameters of our scheme to optimize the mean throughput or η will provide very similar results. However, this is not necessarily the case for other choices of N_i , e.g. when we choose them to minimize the mean completion energy as Figure 10 shows.

Let us turn our attention now to the problem of maximizing the parameter η , i.e. our mean throughput lower bound. Recall that for this setting we are streaming data which is subdivided into blocks that are transmitted them using our scheme. Considering again a satellite link, given a fixed bit error probability ($Pe_{bit} = 0.0001$) let us study the problem of computing the optimal number of bits n per packet given some value of M . In these examples, for the case of a symmetric channel with independent bits $Pe = 1 - (1 - Pe_{bit})^{h+n+gM}$ and $Pe_{ack} = 1 - (1 - Pe_{bit})^{n_{ack}}$.

Figure 11 illustrates the values of η in Mbps given different choices of M and n . First, note that for each value of M there exists an optimal value of n . Thus, an arbitrary choice of n can produce a considerable degradation in performance in terms of throughput. Secondly, there is a (M, n) pair that maximizes the value of η . Finally, the performance of the full duplex network coding and our TDD-T scheme is comparable for different values of n and M .

Figure 12 shows η in Mbps when we change the round-trip time T_{rt} . As expected, a lower T_{rt} allows more throughput in TDD. Again, we observe that our TDD optimal scheme has comparable performance to the full duplex scheme.

Let us compare the performance of our optimal TDD network coding scheme with respect to typical TDD ARQ schemes: Go-back-N (GBN) and Selective Repeat (SR). Fig-

ure 13 shows η for the satellite communications setting with a fixed packet size of $n = 10000$ bits, $n_{ack} = 100$ bits, $T_{rt} = 250$ ms, $Pe_{ack} = 0$ for all schemes, a window size of $W = 10$ for the ARQ schemes, and $g = 20$ bits and $M = 10$ for our network coding scheme. We use different data rates to illustrate different latency scenarios, where higher data rate is related to higher latency. Note that the performance of our scheme is the same as both GBN and SR at low data packet erasure probability, which is expected because the window size W is equal to the block size of our scheme M and we expect very few errors. Our scheme has a slightly lower η for low Pe because each coded data packet includes gM additional bits that carry the random encoding vectors. This effect is less evident as latency increases. In general, our scheme has better performance than GBN.

Figure 13 shows that for low latency (0.1 Mbps) η of our scheme is very close to that of the SR ARQ scheme for all values of Pe , and better than the GBN scheme for high Pe . These results are surprising, because our scheme constitutes a block-by-block transmission scheme which will not start transmission of a new set of M data packets until the previous ones have been received and acknowledged. Note also that, as latency increases, our scheme shows much better performance than the SR scheme for high Pe . The case of 10 Mbps and $Pe = 0.8$ shows that η of our scheme is more than three (3) times greater than that of SR.

Figure 14 shows η for a fixed data rate of 10 Mbps and different T_{rt} . We use a fixed packet size of $n = 10000$ bits, $n_{ack} = 100$ bits, $Pe_{ACK} = 0$ for all schemes, a window size of $W = 10$ for the ARQ schemes, and $g = 20$ bits and $M = 10$ for our network coding scheme. Note that the

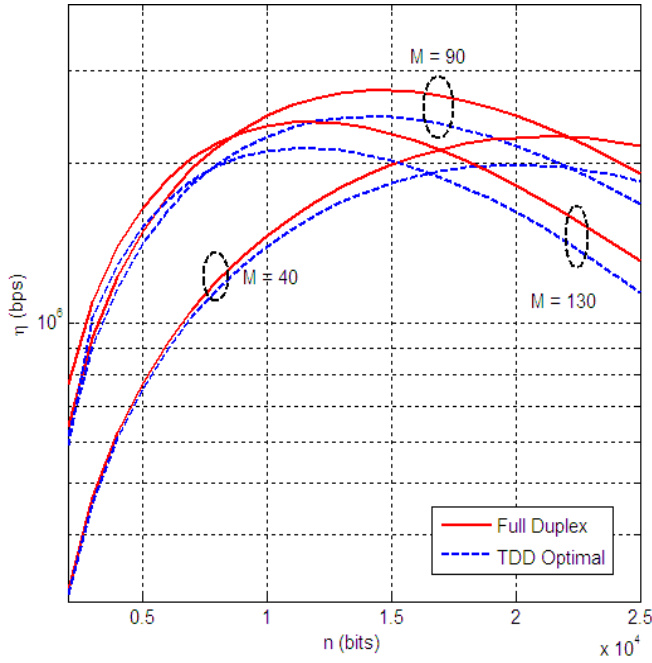


Fig. 11. [2] Throughput measure η versus the number of bits n for the TDD-T and the full duplex schemes in a data packet for a symmetrical channel, for different values of M with parameters $g = 100$ bits, $n_{ack} = 100$ bits, $h = 80$ bits, data rate 100 Mbps, $T_{rt} = 250$ ms, $P_{e_{bit}} = 0.0001$.

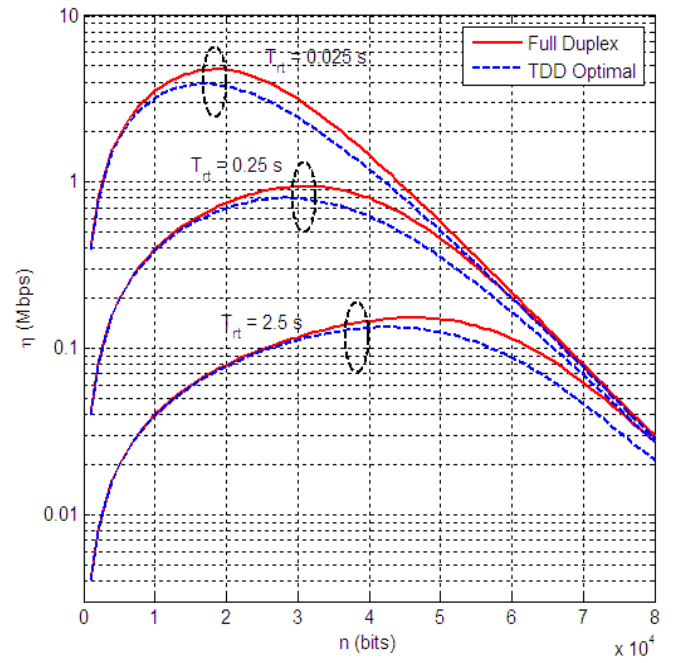


Fig. 12. [2] Throughput η versus n for the TDD-T and the full duplex schemes in a symmetrical channel considering different values of round-trip time T_{rt} with parameters $g = 100$ bits, $n_{ack} = 100$ bits, $h = 80$ bits, data rate 1.5 Mbps, $M = 10$, $P_{e_{bit}} = 0.0001$.

overhead of transmitting M coefficients of g bits per coded packet is only 2%. Thus, this effect cannot be appreciated in the figures. Again, the performance of our scheme is the same as both GBN and SR at low data packet erasure probability. Since the data rate is kept fixed, at higher T_{rt} we get higher latency. The throughput performance is similar to that observed in Figure 13 if we carry our comparison in terms of latency.

Another advantage of our scheme with respect to SR ARQ is that our scheme relies on transmitting successfully one block of M data packets before transmitting a new one. In fact, our scheme minimizes the delay of every block. In contrast, the SR ARQ does not provide any guarantee of delay for any data packet, e.g. the first packet of a file to be transmitted could be the last one to be successfully received. In this sense, our comparison is not completely fair, as it favors the standard schemes. Nonetheless, our scheme is providing similar or better performance than SR but guaranteeing low transmission delays in individual data packets.

IX. CONCLUSION

This paper provides a full characterization of a random linear network coding scheme for reliable communications for time division duplexing channels presented in [2], by providing a recursive expression for the moment generating function. This moment generating function is valid for both the completion time and energy using the appropriate substitutions.

We show that the moment generating function is useful to compute the mean throughput ($MnE[1/T]$) of our scheme. This is achieved by computing the first negative moment of the

completion time, i.e. $E[1/T]$. This metric is different from the typical metric ($Mn/E[T]$) used to characterize the throughput. Numerical results show that both $MnE[1/T]$ and $Mn/E[T]$ are very close when we choose the number of coded data packets to minimize the mean completion time [2]. Thus, optimizing the values of M and n to maximize $Mn/E[T]$ should be very similar to the result we would get using the mean throughput as metric, with the advantage of reducing the required computation.

Although the optimal number of coded data packets, in terms of either mean completion time or mean completion energy, has no closed form solution, we can exploit the recursive characteristic of the problem to simplify our search method. We show that instead of making a M -dimensional search, we can perform M one-dimensional searches to achieve the optimal solution. Finally, these values do not need to be computed in real time. We can reduce the computational load on the nodes by pre-computing these values for different channel conditions and system settings, and storing them as look-up tables in the nodes.

We present means of analysis and numerical results to show that transmitting the optimal number of coded packets before stopping to listen for an ACK is very close to the performance of a full duplex system, while choosing a different number can cause considerable degradation in performance, especially if latency and packet error probability are high.

Also, transmitting the optimal number of coded data packets sent before stopping to listen for an ACK in terms of both mean completion time and energy consumes much less energy in average than a network coding scheme operating in a full

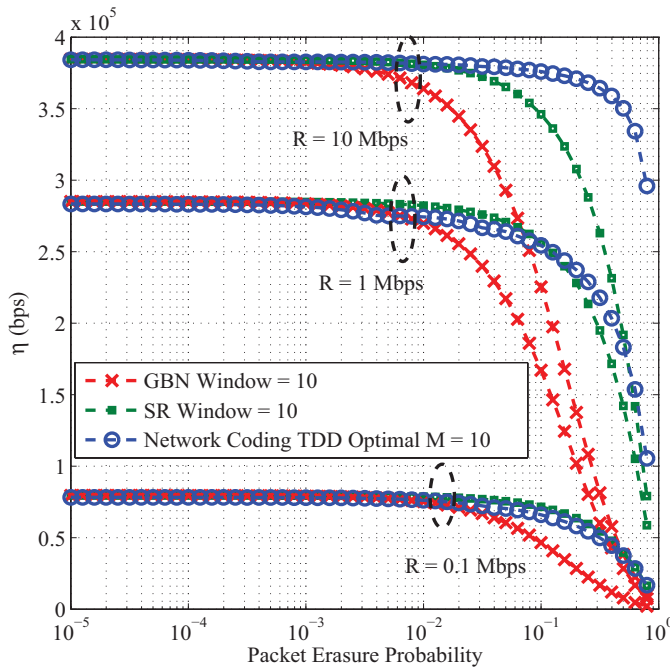


Fig. 13. [2] η versus P_e with two TDD non-network coding schemes (Go-Back-N and Selective Repeat) and our TDD-T network coding scheme, with different R . We used as parameters $g = 20$ bits, $n_{ack} = 100$ bits, $n = 10000$ bits, $h = 80$ bits, $T_{rt} = 250$ ms.

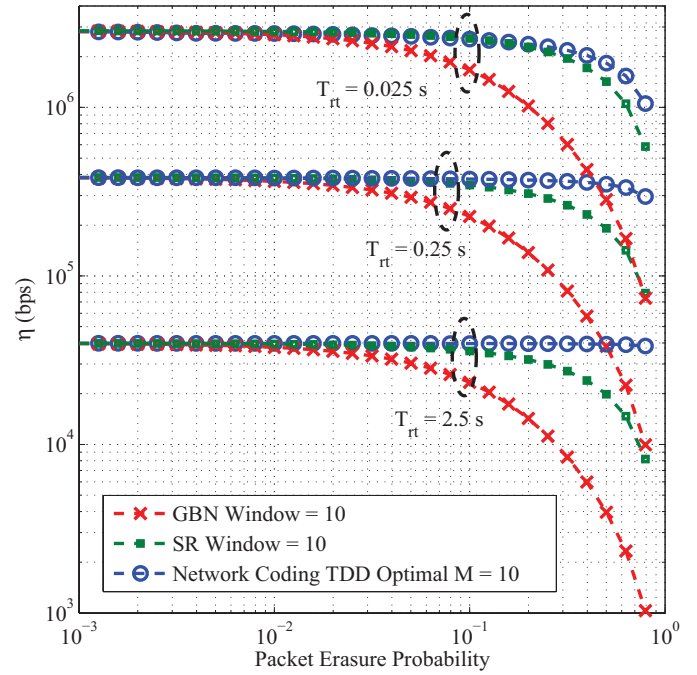


Fig. 14. [2] η versus P_e with two TDD non-network coding schemes (Go-Back-N and Selective Repeat) and our optimal TDD-T network coding scheme, with different T_{rt} values. We used as parameters $g = 20$ bits, $n_{ack} = 100$ bits, $n = 10000$ bits, $h = 80$ bits, $R = 10$ Mbps.

duplex channel. Furthermore, choosing the number of coded data packets to optimize mean completion time, as in [2], provides a good trade-off between energy consumption and completion time.

In terms of throughput performance, we compare our scheme optimized for completion time to the standard half-duplex Go-back-N and Selective Repeat ARQ schemes. Numerical evaluation for different latency shows that our scheme has similar performance to the Selective Repeat in most cases, and considerable performance gain when latencies and packet error probability are high. Numerical results also show that our scheme is superior to Go-back-N when error probability is high for different latency.

Future research will consider an extension of the principles proposed for one link to the general problem of wireless networks, possible due to the use of random network coding. In this extension, each node transmitting through a link, or more generally, a hyperarc (using the terminology in [12]) will have an optimal number of coded packets to transmit.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under grants No. 0520075, 0831728 and CNS-0627021, by ONR MURI Grant No. 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 No. N66001-06-C-2020 (CBMANET).

REFERENCES

- [1] Ahlswede, R., Cai, N., Li, S. Y. R., Yeung, R. W., "Network Information Flow", IEEE Trans. Inf. Theory, vol. 46, no. 4, pp. 1204-1216, Jul. 2000
- [2] Lucani, D. E., Stojanovic, M., Médard, M., "Random Linear Network Coding For Time Division Duplexing: When To Stop Talking And Start Listening", to appear in INFOCOM'09, available at arXiv:0809.2350v1[cs.IT]
- [3] Ozugur, T., Naghshineh, M., Kermani, P., Copeland, J. A., "On the performance of ARQ protocols in infrared networks", Int. Jour. Commun. Syst., vol. 13, pp. 617-638, 2000
- [4] Stojanovic, M., "Optimization of a Data Link Protocol for an Underwater Acoustic Channel", In Proc. Oceans 2005 - Europe, pp. 68- 73, Jun. 2005
- [5] Sastry, A. R. K., "Improving Automatic Repeat-Request (ARQ) Performance on Satellite Channels Under High Error Rate Conditions", IEEE Trans. on Comms., vol. 23, no. 4, pp. 436-439, Apr. 1975
- [6] Akyildiz, I.F., Akan, O.B., Fang, J., "TCP-Planet: A reliable transport protocol for InterPlaNetary Internet", JSAC, vol. 22, no 2, pp. 348361, 2004
- [7] Lucani, D. E., Stojanovic, M., Médard, M., "Random Linear Network Coding For Time Division Duplexing: Energy Analysis", to appear in ICC'09, available at arXiv:0901.0269v1[cs.IT]
- [8] Ho, T., Médard, M., Koetter, R., Karger, D.R., Effros, M., Shi, J., Leong, B., "A Random Linear Network Coding Approach to Multicast", Trans. Info. Theory, vol. 52, no. 10, pp.4413-4430, Oct. 2006
- [9] Chao, M. T., Strawderman, W. E., "Negative Moments of Positive Random Variables", Jour. of the American Statistical Asso., vol. 67, no. 338, pp. 429-431, Jun. 1972
- [10] Cressie, N., Davis, A. S., Folks, J. L., Policello II, G. E., "The Moment-Generating Function and Negative Integer Moments", The American Statistician, vol. 35, no. 3, pp. 148-150, Aug. 1981
- [11] Chapeau-Blondeau, F., and Monir, A., "Numerical Evaluation of the Lambert W Function and Application to Generation of Generalized Gaussian Noise With Exponent 1/2", IEEE Trans. on Signal Proc., Vol. 50, No. 9, Sept. 2002
- [12] Lun, D. S., Ratnakar, N., Médard, M., Koetter, R., Karger, D. R., Ho, T., Ahmed, E., Zhao, F., "Minimum-Cost Multicast Over Coded Packet Networks", IEEE Trans. on Info. Theory, vol. 52, no. 6, pp.2608-2623, Jun.2006