

Network Codes with Deadlines

Elona Erez[†] Michelle Effros^{††} and Tracey Ho^{††}

Abstract—We investigate the effect of decoding deadlines on network coding capacity, specifically, the capacity curve of a network as a function of the allowed delay from the time a set of bits is transmitted by the source to the time it is fully decoded by the sinks. We show that scalar linear codes are not optimal even for multicast when the data has deadlines. In fact, infinite blocklength is required in general in order to achieve the optimal performance of linear block codes in these scenarios. We study the case of two types of data, where the first type has a tighter deadline than the other. We find for an interesting family of networks the optimal linear convolutional codes. We formulate a code design criterion for general networks with two data types. Finally, as an alternative approach, we show that the problem of multicast with deadlines can be transformed into a non-multicast problem without deadlines in an extended network. Using that approach, we find an upper bound on the complexity of checking the feasibility of the problem.

I. INTRODUCTION

We consider communications applications where each packet is labeled with a deadline and is required to be reconstructed at the receiving end before that deadline. The packets expire and become useless after their deadline. One of the challenges is to determine the possible rates achieved in this scenario. In other words, we seek to characterize the capacity curve of a network, as a function of the packet deadlines. The rates achieved by network coding for multicast without deadlines are well understood. A multicast contains a single source, s , and d receivers, t_1, \dots, t_d ; all sinks need to receive all of the information. The smallest of the minimal cuts between the source and any of the individual sinks can be achieved [1]. The optimal rate can be achieved with linear network codes [2] if the size of the algebraic field from which the coding coefficients are taken is large enough [3].

Yet there are some aspects in this problem which are still not well understood. While the number of sinks is a sufficient alphabet size for multicast network coding, it is not known in general what is the *necessary* field size. It was shown in [4],[5],[6] that an alphabet size of \sqrt{d} is necessary for some networks. However, for a given network, it is not clear how the minimal necessary alphabet size can be found.

The problem of the minimal alphabet size is related to our problem of deadlines because for block network codes, where it is assumed that the coding is over symbols from an

algebraic field, the delay can be estimated from the logarithm of the alphabet size (i.e., the blocklength). If we consider more general codes, in which, for example, the field size of the coefficients are allowed to vary from one node to the next in the network or the elements of the code form a ring instead of a field, such as in the case of convolutional codes, then the problem becomes even more complicated.

In this paper we design and analyze the structure of optimal codes for networks with deadlines. We choose a simple example and use it in order to prove several interesting results about capacity for networks with deadlines. First, we show that linear algebraic scalar codes do not achieve the optimal rate of linear network codes with deadlines. Furthermore, we show that any finite blocklength linear code cannot in general achieve the optimal rate-deadline performance. Specifically, we show that when the source is required to transmit data at the multicast capacity of the network, linear convolutional codes which have infinite blocklength achieve a strictly smaller deadline than any linear code with finite blocklength. Additionally, we show that for that simple example, the rate-deadline performance of convolutional codes is in fact optimal. This might indicate the convolutional codes are more adequate for networks with deadline.

While the majority of the paper focuses on the case where the deadline for each packet is a fixed d times steps after that packet's arrival, we generalize our approach also to networks with multiple packet types. We focus on the case of two data types, with deadlines d_1 and d_2 after their respective arrival times. For this scenario, we find the optimal linear convolutional code for our example network. We formulate a design criterion for general networks with two data types.

As an alternative approach to the problem, we show that multicast with deadlines can be represented as a non-multicast problem without deadlines. Using this approach, we find an upper bound on the complexity of verifying the feasibility of the problem.

The outline of the paper is as follows. In Section II we introduce some notation and definitions. In Section III we show that for some networks, the optimal deadline-capacity curve cannot be achieved by algebraic scalar codes. In Section IV, we show that for some networks the optimal deadline cannot be achieved by any code with finite blocklength. In Section V, we consider the case where there are two data types. In Section VI, we show as an alternative approach to the problem, that the multicast problem with deadlines can be represented by an equivalent non-multicast problem without deadlines. In Section VII, we conclude the paper.

[†] The research was performed while E. Erez was with the Department of Electrical Engineering, California Institute of Technology. She is now with the Department of Electrical Engineering, Yale University, New Haven, CT, 06511, USA, elona.erez@yale.edu

^{††} California Institute of Technology, Pasadena, CA, 91125, USA, {tho, effros}@caltech.edu.

This work has been supported in part by the Defense Advanced Research Projects Agency (DARPA) under Contract No. W911NF-07-1-0029, and by Caltech's Lee Center for Advanced Networking.

II. NOTATION AND DEFINITIONS

Consider an acyclic, unit capacity directed network $G = (V, E)$ where parallel edges are allowed. There is a single source s and a set of d sinks $T = \{t_1, \dots, t_d\} \subset V$. The data rate of source s is R . Denote by h the size of the minimal individual min-cut between s and any of the sinks. Denote by $\Gamma_I(v)$ and $\Gamma_O(v)$ the set of incoming and outgoing edges of node v , respectively. For the directed edge (v, v') from v to v' , v is the tail of e and v' is the head of e . Denote by $\Gamma_I(e)$ and $\Gamma_O(e)$ the set of incoming edges of the tail of e and the set of outgoing edges of the head of e , respectively.

For *linear network codes*, any edge e has a global coding vector $\mathbf{v}(e)$ of dimension R associated with it. For *linear algebraic block network codes* the source s gets input symbols denoted by $X = (X_1, \dots, X_R)$ from the field \mathcal{F} . The global coding vector $\mathbf{v}(e)$ is given recursively by

$$\mathbf{v}(e) = \sum_{e' \in \Gamma_I(e)} m(e', e) \mathbf{v}(e'), \quad (1)$$

where $m(e', e)$ is the *coding coefficient* for edges e' and e . Let $y(e)$ be the symbol transmitted on an edge e , then

$$y(e) = \sum_{e' \in \Gamma_I(e)} m(e', e) y(e') = \mathbf{v}(e)^T X. \quad (2)$$

For *linear convolutional network codes* over the binary field, let $F(D)$ denote the ring of polynomials over the binary field with variable D . The variable D represents a unit time shift. Each element of $\mathbf{v}(e)$ is an element of $F(D)$. The source node s gets R input binary streams $x_1(n), \dots, x_R(n)$. The source starts transmitting bits at time $n = 0$. The power series in variable D of the input stream $x_j(n)$ is

$$X_j(D) = \sum_{n=0}^{\infty} x_j(n) D^n. \quad (3)$$

The global coding vector associated with edge e is given by the same relation as in (1), except all of the elements are taken from $F(D)$ and the additions and multiplications are regular for polynomials in $F(D)$.

Let $y_e(n)$ be the symbol transmitted on edge e at time instant n ; then the power series in variable D is

$$\begin{aligned} Y_e(D) &= \sum_{n=0}^{\infty} y_e(n) D^n = \sum_{e' \in \Gamma_I(e)} m(e', e) Y_{e'}(D) \\ &= \mathbf{v}(e)^T \mathbf{x}(D), \end{aligned} \quad (4)$$

where $\mathbf{x}(D) = (X_1(D), \dots, X_h(D))^T$.

Unless stated otherwise, we treat in this paper bits as the most basic units of information. The delay of our model is independent of the link delays, in order to eliminate the effect of the network itself on the delay, and to consider only the effect of the network code on the delay. We therefore assume that all the links in the network have zero delay. The delay in our model is the decoding delay. At a certain instant, the source transmits a set of R bits (or $\lceil R \rceil$ if R is not an integer), where we recall that R is the rate of the source. Since there are no link delays, after the transmission the sinks

are assumed to instantly receive bits at their incoming links. However, the sinks might not be able to instantly decode the set of R bits. The source might be required to transmit additional bits, in order for the sinks to be able to decode the original set of R bits. The number of time steps until all the sinks are able to decode the set of R bits is the *bits delay*. In other words:

Definition 1: Denote by t_1 the first instant when the source has finished transmitting a set of $\lceil R \rceil$ bits. Denote by t_2 the first instant when the same set of $\lceil R \rceil$ bits are fully decodable by the last sink in the network. The *bits delay* is:

$$\delta = t_2 - t_1 \quad (5)$$

Definition 2: The *bits deadline* Δ is the largest allowed bits delay.

III. SCALAR ALGEBRAIC NETWORK CODES ARE NOT DEADLINE OPTIMAL

Scalar algebraic network codes are “one shot” in the sense the once the symbols, which are elements of an algebraic field, are transmitted by the source and reach the sink they can be decoded, without the need to wait for future symbols.

Consider the family of networks represented by Figure 1. All edges have unit capacity. The source s is connected to

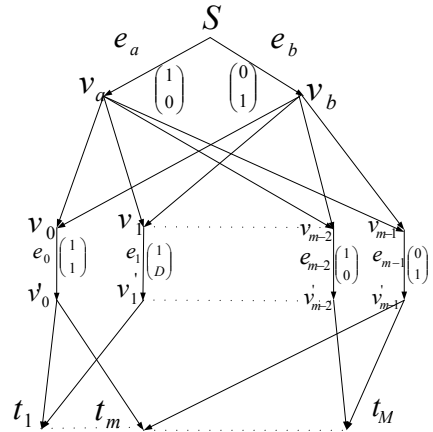


Fig. 1. Example of a Network for Deadline Analysis

nodes v_a and v_b . Both nodes v_a and v_b are connected to all nodes v_0, \dots, v_{m-1} . Each node v_i is connected only to node v'_i . For each pair of nodes from $\{v'_0, \dots, v'_{m-1}\}$, there is a sink t_j connected to only those nodes. Thus there are $d = \binom{m}{2}$ sinks. Denote by e_i the edge between v_i and v'_i . An equivalent network is called an $\binom{m}{2}$ combination network in [7]. The capacity of this network without deadlines is 2.

For scalar algebraic network codes the alphabet size required for this network for capacity 2 is at least $m - 1$, as shown in [4],[5],[6]. This is because the global vectors of edges e_0, \dots, e_{m-1} must together form an MDS (Maximum Distance Separable) code with the parameters $n = m, k = 2$. For these parameters, the minimal field size for any algebraic code, with any field size, is at least $m - 1$.

In this section, we assume that all bits have the same deadline. For this case, it is reasonable to assume that there

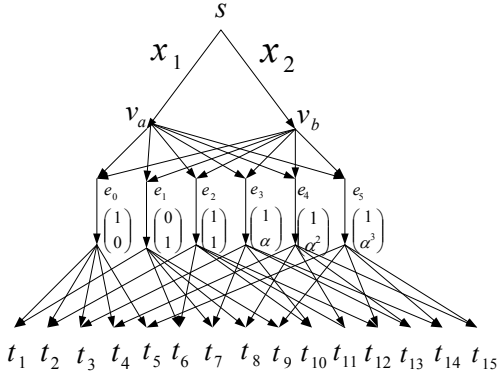


Fig. 2. Example network with a scalar algebraic network code

are only integer deadlines. We assume that if the deadline is fractional it is rounded to the next smaller integer.

Consider the $\binom{6}{2}$ combination network in Figure 2. For this network, we find the capacity region of algebraic scalar network codes. We then find the capacity region of a specific time-variant linear block code and show that its rate region improves over that of scalar codes. This shows that algebraic scalar codes do not achieve in general the optimal capacity-deadline performance. Furthermore, we show that for this network, a convolutional linear code achieves the optimal performance. This might indicate that convolutional codes are more appropriate for networks with deadlines.

Lemma 1: The rate region of scalar algebraic codes of the $\binom{6}{2}$ combination network is given by the dashed line in Figure 3.

Proof: Recall that we assume that the deadline is an integer. We therefore analyze the deadlines $\Delta = 0, 1, 2$. We begin by showing that for $\Delta = 2$ the capacity $h = 2$ is achievable using scalar algebraic network codes. We will later analyze the cases $\Delta = 0, 1$.

For scalar algebraic network codes the alphabet size required to achieve capacity $h = 2$ is at least $m - 1 = 5$. Thus, the blocklength of the symbol is $\log_2 5$. Since bits are our most basic units, the blocklength is in fact $\lceil \log_2 5 \rceil = 3$. The source transmits the first R bits at time $t_1 = 0$. The first bit of each symbol is received by the sinks at time $n = 0$, the second bit at time $n = 1$, and the last bit of the symbol at time $n = 2$. Thus all the sinks are able to decode the symbols at the end of the block which is at time $t_2 = \lceil \log_2 5 \rceil - 1 = 2$. Thus, the delay of the algebraic code according to Definition 1 is $t_2 - t_1 = 2$. Figure 2 shows the global coding vectors of e_0, \dots, e_5 for a code that achieves deadline 2, where α is the solution of the equation $x^3 + x + 1 = 0$. This code achieves the capacity $h = 2$ of the network with deadline $\Delta = 2$. A better rate cannot be achieved since this is the capacity of the network without deadlines.

If $\Delta = 1$ or smaller then rate 2 cannot be achieved with scalar codes because otherwise it will be a contradiction to the MDS bound for $n = m, k = 2$ [4],[5],[6]. Since our basic units are bits, for scalar codes the possible rates are also integers. Therefore, for $\Delta = 1$ the optimal rate is 1, achieved by simple transmission of a bit throughout the network, without coding. Likewise for $\Delta = 0$, the optimal

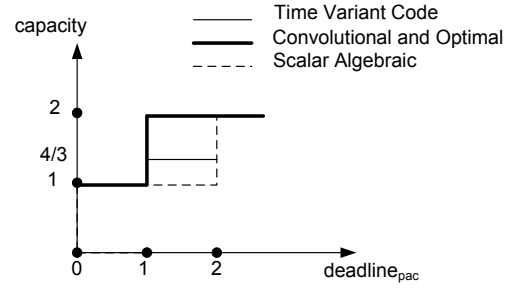


Fig. 3. Rate region for example network

rate cannot be larger than for $\Delta = 1$, which is 1. Rate 1 can be achieved also for $\Delta = 0$ using simple transmission. ■

Linear time-variant codes can be represented by global coding vectors as in (1), except the global coding vector $\mathbf{v}_e(n)$ is allowed to vary from one time step n to the next.

Lemma 2: The capacity region of time-variant linear block codes of the $\binom{6}{2}$ combination network contains the rate region marked by the solid thin line in Figure 3.

Proof: For $\Delta = 0$ rate 1 can be achieved using simple transmission without deadline. For $\Delta = 2$, we have seen in Lemma 1 that scalar algebraic codes achieve rate 2. Since scalar algebraic codes are a special case of time-variant codes, rate 2 is achieved also by time-variant codes.

For $\Delta = 1$, we present a time-variant linear code, which achieves the rate $R = 4/3$. In our notation, $x_i(j)$ designates the j th symbol of source x_i . Source s transmits to v_a the symbol $x_1(0)$ at time $n = 0$ and $x_1(1)$ at time $n = 1$. The source s transmits to v_b the symbol $x_2(0)$ at time $n = 0$ and $x_2(1)$ at time $n = 1$. The bits received at the heads of e_0, \dots, e_5 at times $n = 0, 1, 2$ are shown in Table I. For $n = 0$ the code at the heads of e_0, e_1, e_2 is a binary MDS code, and the code at the heads of e_3, e_4, e_5 is the same MDS code. For $n = 1$ the code at the heads of edges e_0, e_1, e_2 is an MDS code and the code at the heads of e_3, e_4, e_5 complements what was not received on these edges at time $n = 0$. For $n = 2$ the code at the heads of e_0, e_1, e_2 complements what was not received at these edges at time $n = 1$, and the code at the heads of e_3, e_4, e_5 is an MDS code. After 3 time steps all sinks are able to decode 4 bits. Thus, the capacity of the code is $R = 4/3$, and $\lceil R \rceil = 2$. The source transmitted the bits $x_1(0), x_2(0)$ at time $n = 0$ and $x_1(1), x_2(1)$ at time $n = 1$. All sinks can decode $x_1(0), x_2(0)$ at time $n = 1$ and $x_1(1), x_2(1)$ at time $n = 2$. It follows from definition 2 that the deadline of this scheme is 1. Since the decoder cannot decode in a “one shot” fashion, the code is not scalar. The rate region of the time-variant code is indeed represented by the thin solid line in Figure 3. ■

For the same deadline, we have seen that scalar algebraic codes can achieve only rate $R = 1$. We note that the rate region of the time-variant code in Figure 3 represents an achievable rate region for time-variant codes, and might be even further improved by another time-variant code with a finite blocklength. We do not consider here such other codes. However, Lemma 1 together with Lemma 2 suffice in order

TABLE I
AN EXAMPLE OF A TIME VARIANT CODE (RECEIVED SYMBOLS)

n	e_0	e_1	e_2	e_3	e_4	e_5
0	$x_1(0)$	$x_2(0)$	$x_1(0) + x_2(0)$	$x_1(0)$	$x_2(0)$	$x_1(0) + x_2(0)$
1	$x_1(1)$	$x_2(1)$	$x_1(1) + x_2(1)$	$x_2(0)$	$x_1(0)$	$x_1(0)$
2	$x_2(1)$	$x_1(1)$	$x_1(1)$	$x_1(1)$	$x_2(1)$	$x_1(1) + x_2(1)$

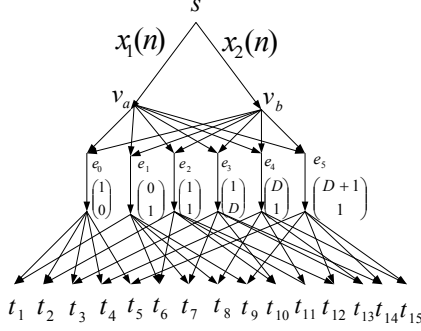


Fig. 4. Convolutional code for example network

to show the following theorem:

Theorem 1: In general, for a given deadline, linear scalar algebraic network codes do not achieve the optimal rate of linear codes with finite blocklength.

Proof: Immediate from Lemma 1 and Lemma 2. ■

This result is surprising since, in contrast, scalar algebraic network codes achieve the optimal rate for the case of no deadline. We turn now our attention to convolutional codes with an infinite blocklength and show the following theorem:

Theorem 2: Linear convolutional network codes achieve the optimal rate region of the $\binom{6}{2}$ combination network.

Proof: We find in the following a convolutional code that achieves rate 2 for $\Delta = 1$. Consider the code in Figure 4. The symbols on the heads of e_0, \dots, e_5 at times $n = 0, 1, 2, 3$ are shown in Table II. We again use the notation $x_i(j)$ to designate the j th symbol of source x_i . All sinks can decode $x_1(0), x_2(0)$ at time $n = 1$. Likewise, all other bits are decoded after a delay of a single time step. Therefore, $\delta = 1$ and $\Delta = 1$ for rate $R = 2$.

Alternatively, the delay of convolutional network codes can be found using the approach in [8],[9],[10]. Denote by $A_l(D)$ the matrix that contains the global coding vectors of the edges in $\Gamma_l(t_l)$ as its rows. That is, for sink t_l connected to edges e_i and e_j , $A_l(D)$ maintains the relation

$$\begin{aligned} A_l(D)\mathbf{x}(D) &= \begin{pmatrix} \mathbf{v}^T(e_i) \\ \mathbf{v}^T(e_j) \end{pmatrix} \begin{pmatrix} X_1(D) \\ X_2(D) \end{pmatrix} \\ &= \begin{pmatrix} Y_i(D) \\ Y_j(D) \end{pmatrix} \end{aligned} \quad (6)$$

where $Y_i(D), Y_j(D)$ are the power series of the bits received at edges e_i, e_j , respectively. For example, for t_{14} which is connected to e_3 and e_5 the matrix $A_{14}(D)$ is given by

$$A_{14}(D) = \begin{pmatrix} \mathbf{v}^T(e_3) \\ \mathbf{v}^T(e_5) \end{pmatrix} = \begin{pmatrix} 1 & D \\ D+1 & 1 \end{pmatrix} \quad (7)$$

It was shown in [8],[9],[10] that the delay can be found

from $\det A_l(D)$, the determinant of $A_l(D)$. The determinant is a polynomial in variable D . If the power of the smallest degree additive term of the polynomial $\det A(D)$ is N , then the bits delay is also N . For sink t_{14} , connected to e_3 and e_5 , the determinant $\det A_l(D)$ equals $D^2 + D + 1$. The smallest degree additive term is 1 and its power is $N = 0$. Thus the bits delay is 0. For each $l \in 1, \dots, 15$, $\det A_l(D)$ is given respectively, by: 1, 1, D , 1, 1, 1, 1, D , $D+1$, $D+1$, $D+1$, D , $D^2 + 1$, $D^2 + D + 1$, 1. Therefore, the bits delays for t_1, \dots, t_{15} are, respectively, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, and deadline 1 can be achieved. The capacity of this scheme is 2 since according to [8],[9] the code can be decoded using a sequential decoder while achieving the capacity of the network. The capacity is achieved asymptotically because at the first time instant some sinks do not receive rate 2. But subsequently, all sinks can decode 2 bits at each time step. Thus the convolutional code achieves asymptotically the optimal rate, since the min-cut between the source and each sink is 2.

For deadline $\Delta = 0$, convolutional codes can achieve rate 1, again by simple transmission without coding. For deadline $\Delta = 0$, rate 1 is also an upper bound for the optimal scheme. This is because, as mentioned above, bits are our most basic unit. If 1 is not an upper bound, then all of the sinks would be able to decode at least 2 symbols after a single time step. This contradicts the MDS bound, since there are no binary MDS codes of length 6 with dimension $k = 2$ [11]. We cannot even asymptotically achieve anything higher than rate 1. A deadline of $\Delta = 0$ requires that the source bits must be decoded by all sinks in one time step, so by the MDS bound the source can only send a single bit on both outgoing edges. By similar reasoning, in each time step the source must send a single bit on both outgoing edges. ■

Corollary 1: For the $\binom{3}{2}$ combination network, the capacity is 2 for deadline $\Delta \geq 0$. For the $\binom{4}{2}$ and $\binom{5}{2}$ combination networks, the capacity is 1 for $0 \leq \Delta < 1$ and 2 for $\Delta \geq 1$. For these 3 networks, rate 1 can be achieved without coding. Rate 2 can be achieved with a $(k = 2, n = m)$ MDS scalar algebraic linear network code with blocklength $\lceil \log_2(m-1) \rceil$ (where $m = 3, 4$ or 5, respectively).

IV. FINITE BLOCK-LENGTH CAPACITY ACHIEVING CODES

In this section we discuss linear block codes that achieve the multicast capacity, which is the minimal of the individual min-cuts between the source and any of the sinks. We focus on block codes over the binary field, since a bit is our most basic unit. The codes are allowed to be time variant. Of all these codes, we call the code that achieves the minimal

TABLE II
AN EXAMPLE OF A CONVOLUTIONAL CODE (RECEIVED SYMBOLS)

n	e_0	e_1	e_2	e_3	e_4	e_5
0	$x_1(0)$	$x_2(0)$	$x_1(0) + x_2(0)$	$x_1(0)$	$x_2(0)$	$x_1(0) + x_2(0)$
1	$x_1(1)$	$x_2(1)$	$x_1(1) + x_2(1)$	$x_1(1) + x_2(0)$	$x_1(0) + x_2(1)$	$x_1(0) + x_1(1) + x_2(1)$
2	$x_1(2)$	$x_2(2)$	$x_1(2) + x_2(2)$	$x_1(2) + x_2(1)$	$x_1(1) + x_2(2)$	$x_1(1) + x_1(2) + x_2(2)$
3	$x_1(3)$	$x_2(3)$	$x_1(3) + x_2(3)$	$x_1(3) + x_2(2)$	$x_1(2) + x_2(3)$	$x_1(2) + x_1(3) + x_2(3)$

deadline according to Definition 1, *the optimal deadline code*. In this section we show that for some networks, the optimal deadline code necessarily has infinite blocklength. The infinite blocklength code achieves the capacity rate only asymptotically, since the sinks may not receive the full capacity rate in the first few transmissions. After these few transmissions, the sinks all receive the rate equal to the capacity. Since the transmission is infinite, the capacity is achieved asymptotically. Before formally stating and proving this result, a few definitions are required. Since we do not discuss noisy channels in this paper we use the term "noise" in the following definitions to describe any component of the source signal that is not currently required to be decoded.

Definition 3: Define the time-variant binary global vector $\mathbf{v}_e(n)$:

$$y_e(n) = \mathbf{v}_e(n)^T X, \quad (8)$$

where X is the binary representation of the block source, $y_e(n)$ is the received bit at time n at edge e and $\mathbf{v}_e(n)$ represents the linear relation between the transmitted block and the received bit at time n .

Note that time-invariant block codes over the algebraic field \mathcal{F} of size 2^m for some integer m that are represented by equation (2) are a special case of the time-variant codes that can be represented by the relation (8).

Definition 4: The total rank received by sink t is the rank of the matrix $A(t)$, whose rows are $\mathbf{v}_e(n)^T$ for all incoming edges e of the sink t and all n for which the vector $\mathbf{v}_e(n)^T$ is not the zero vector. The rank received at the sink during the time period $i \leq n \leq j$ is the rank of the submatrix of $A(t)$ whose rows are $\mathbf{v}_e(n)^T$ for all incoming edges e of the sink t and all n in the time range $i \leq n \leq j$.

Definition 5: We split the vector X into two vectors X^S and X^N in concatenation

$$X = \begin{pmatrix} X^S \\ X^N \end{pmatrix} \quad (9)$$

where the dimension of X^S is denoted by L_S and the dimension of X^N is denoted by L_N . The sum $L_S + L_N$ is equal to the dimension of X . We call X^S the signal component and X^N the noise component. For each edge e we define the vector \mathbf{v}_e^S and \mathbf{v}_e^N such that

$$y_e(n) = \mathbf{v}_e^S(n)^T X^S + \mathbf{v}_e^N(n)^T X^N. \quad (10)$$

Definition 6: The signal rank received by sink t is the rank of the matrix $A^S(t)$, whose rows are $\mathbf{v}_e^S(n)^T$ for all incoming edges e of the sink t and all n for which the vector $\mathbf{v}_e^S(n)^T$ is not the zero vector. The signal rank received at the sink

during the time period $i \leq n \leq j$ is the rank of the submatrix of $A^S(t)$ whose rows are $\mathbf{v}_e^S(n)^T$ for all incoming edges e of the sink t and all n in the time range $i \leq n \leq j$.

Definition 7: The noise rank received by sink t is the rank of the matrix $A^N(t)$, whose rows are $\mathbf{v}_e^N(n)^T$ for all incoming edges e of the sink t and all n for which the vector $\mathbf{v}_e^N(n)^T$ is not the zero vector. The noise rank received at the sink during the time period $i \leq n \leq j$ is the rank of the submatrix of $A^N(t)$ whose rows are $\mathbf{v}_e^N(n)^T$ for all incoming edges e of the sink t and all n in the time range $i \leq n \leq j$.

Theorem 3: For some multicast networks, the capacity-achieving linear code that achieves the minimal deadline is required to have infinite blocklength. The capacity rate is achieved in this case only asymptotically.

Proof: Assume that for the network in Figure 2 there exists a block code with finite blocklength that achieves the multicast capacity rate, which for this network is 2. We denote the blocklength by N_f . We show by contradiction that no such code achieves the same deadline achieved by the convolutional codes shown in Figure 4, which is $\Delta = 1$. Assume that the block code achieves the same deadline achieved by the convolutional codes shown in Figure 4, which is 1. Thus, by the time instant $n = 1$ (that is, after the sink connected to edges e_i and e_j receives two bits from e_i and two bits from e_j), all the sinks are required to decode two bits, both x_1 and x_2 . We define the signal component from Definition 5 as $X^S = (x_1, x_2)^T$ and the noise component as $X^N = (x_3, \dots, N_f)^T$ with $L_S = 2$ and $L_N = N_f - 2$. The vectors $\mathbf{v}_e^S(n)$ and $\mathbf{v}_e^N(n)$ are then found using (10).

Consider the symbols received at the sink connected to e_i, e_j at time instants $n = 0, 1$. Denote the symbol of edge e_i at time n by $y_i(n)$. Then from (10)

$$y_i(0) = \mathbf{v}_e^S(0)^T X^S + \mathbf{v}_e^N(0)^T X^N \quad (11)$$

$$= \mathbf{v}_e^S(0)^T (x_1, x_2)^T + \mathbf{v}_e^N(0)^T (x_3, \dots, N_f)^T$$

$$y_i(1) = \mathbf{v}_e^S(1)^T X^S + \mathbf{v}_e^N(1)^T X^N \quad (12)$$

$$= \mathbf{v}_e^S(1)^T (x_1, x_2)^T + \mathbf{v}_e^N(1)^T (x_3, \dots, N_f)^T$$

At each instant n , the total rank that each sink receives according to Definition 4 (with $i = n = j$) must be exactly 2. Otherwise, the multicast capacity of 2 would not be achieved. Similarly, to achieve the capacity, the total rank received by each sink over the time period $0 \leq n \leq 1$ must be 4.

Consider the matrix A of size $12 \times N_f$ whose rows are $\mathbf{v}_{e_i}(n)^T$ for all $e_i, 1 \leq i \leq 6$ and for all $0 \leq n \leq 1$. As we show in the following, the rank of matrix A must be at least 5. Since as noted above, the total rank received by each

sink over the time period $0 \leq n \leq 1$ is 4, it follows that the rank of A is at least 4. If the rank of A is *exactly* 4, then we have found a multicast block code that achieves the capacity 2 with blocklength 2. That contradicts the MDS bound for codes with blocklength 6 and dimension $k = 2$ [4],[5],[6]. It follows that indeed the rank of matrix A is at least 5.

Next consider the matrix A^S of size $12 \times N_f$ whose rows are $\mathbf{v}_{e_i}^S(n)^T$ for all $e_i, 1 \leq i \leq 6$ and for all $0 \leq n \leq 1$. The rank of A^S must be exactly 2, since otherwise the sinks will not be able to decode both x_1 and x_2 by time $n = 1$. Further, consider the matrix A^N of size $12 \times N_f$ whose rows are $\mathbf{v}_{e_i}^N(n)^T$ for all $e_i, 1 \leq i \leq 6$ and for all $0 \leq n \leq 1$. Since $A = A^S + A^N$, the rank of A^S is 2, and the rank of A is at least 5, it follows that the rank of A^N is at least 3.

Finally, define for each $e_i, 1 \leq i \leq 6$ the matrix A_i^N whose rows are $\mathbf{v}_{e_i}^N(0)^T$ and $\mathbf{v}_{e_i}^N(1)^T$. Assume that there is an edge e_i for which the rank of the matrix A_i^N is 2. Since the rank of A^N is at least 3 there exists at least one $j \neq i$ such that the rank of the matrix whose rows are $\mathbf{v}_{e_i}^N(0)^T, \mathbf{v}_{e_i}^N(1)^T, \mathbf{v}_{e_j}^N(0)^T, \mathbf{v}_{e_j}^N(1)^T$ is at least 3. The sink connected to e_i and e_j would not be able to decode the signal x_1, x_2 by the time $n = 1$ since the noise it receives has rank 3. It follows that the deadline would not be achieved.

It follows that for each i the rank of A_i^N is either 0 or 1. Assume there is an edge e_i for which the rank of A_i^N is 0. A sink connected to e_i and e_j (for $j \neq i$) will receive noise of rank at most 1 (the rank of A_j^N) during the time period $0 \leq n \leq 1$. Therefore, the total rank that the sink receives (signal and noise) during $0 \leq n \leq 1$ is at most 3. Thus, the code would not achieve the capacity, which is 2.

We conclude that for each i the rank of the matrix A_i^N is exactly 1. For each i there are three possibilities: (1) $\mathbf{v}_{e_i}^N(0)^T = \mathbf{v}_{e_i}^N(1)^T$, (2) $\mathbf{v}_{e_i}^N(0)^T = 0$, (3) $\mathbf{v}_{e_i}^N(1)^T = 0$. Since for each i we can change the order of the symbols received at $n = 0$ and $n = 1$ without changing the deadline properties of the code, we assume without loss of generality that the two possibilities are: (1) $\mathbf{v}_{e_i}^N(0)^T = \mathbf{v}_{e_i}^N(1)^T$ (2) $\mathbf{v}_{e_i}^N(1)^T = 0$. Since the code is linear, we can always reduce case (1) to case (2). In case (1) the decoder can add the two symbols it receives without rank loss and use the symbols $y_i(0), y_i(0) + y_i(1)$ for decoding instead of $y_i(0), y_i(1)$. For this equivalent code the resulting $\mathbf{v}_{e_i}^N(1)^T$ is the zero vector.

Since $\mathbf{v}_{e_i}^N(1)^T = 0$ for each i , it follows that $\mathbf{v}_{e_i}(1) = \mathbf{v}_{e_i}^S(1)$. Since the code is binary, there are only three possibilities $y_i(1) \in \{x_1, x_2, x_1 + x_2\}$ for each i . But we have 6 edges e_i , and the symbols $y_i(1), i = 1, \dots, 6$ must be mutually independent in order for all sinks to receive full rank. Clearly, this property cannot be achieved.

We conclude that the capacity $h = 2$ with deadline $\Delta = 1$ cannot be achieved by codes with finite blocklength. Since the convolutional code (with infinite blocklength) shown in Figure 4 achieves asymptotically the capacity $h = 2$ with deadline $\Delta = 1$ the theorem follows. ■

V. TWO DATA TYPES

We next consider networks carrying multiple data types with distinct deadline constraints. For example, when file downloads and video traverse the same network, the deadline on each video packet is likely to be tighter than the deadline of an individual file packet. The deadline of the more time-sensitive data type is denoted by d_1 and the deadline of the less time-sensitive data type is denoted by d_2 , giving $d_1 < d_2$. In Section V-A, we consider the case where $d_1 = 0$ is fixed and design for the $\binom{m}{2}$ combination network the linear convolutional code that minimizes d_2 subject to the constraint on d_1 and the code type. In Section V-B we consider general criteria for general networks carrying two data types with distinct deadlines.

A. Code Design for a Family of Networks

Theorem 4: Consider the convolutional code in Figure 1. The global coding vectors $\mathbf{v}(e)$ are

$$\begin{aligned} \mathbf{v}(e_a) &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{v}(e_b) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ \mathbf{v}(e_i) &= \begin{pmatrix} 1 \\ D^i \end{pmatrix}, i = 0, \dots, m-3, \\ \mathbf{v}(e_{m-2}) &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{v}(e_{m-1}) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \end{aligned} \quad (13)$$

For a given deadline $d_1 = 0$, this code achieves the smallest deadline d_2 of all linear convolutional code with rate 2.

Proof: We consider the structural properties that any linear convolutional code with $d_1 = 0$ and rate 2 must have. We show that for any code that has these properties, the deadline d_2 must be at least as large as that of the code in the theorem. For any linear convolutional network code the elements of the vector $\mathbf{v}(e_i)$ are polynomials in D , possibly the zero polynomial. For convenience of presentation, we define the zero polynomial as the polynomial D^∞ .

If the power of the smallest degree additive term of the first element of $\mathbf{v}(e_i)$ is larger than 0, then the symbol transmitted at $n = 0$ on e_i will not depend on $x_1(0)$; this symbol will reach the sink at $n = 0$. Consider the first element of the two vectors $\mathbf{v}(e_i), \mathbf{v}(e_j)$ for some i, j such that $0 \leq i, j \leq m-1, i \neq j$. If for the first element of both $\mathbf{v}(e_i)$ and $\mathbf{v}(e_j)$ the power of the smallest degree additive term is larger than 0, then the sink connected to e_i and e_j will not be able to decode $x_1(0)$ by $n = 0$ and the deadline requirement will not be met. It follows that at least one of the vectors $\mathbf{v}(e_i)$ or $\mathbf{v}(e_j)$ will have as the first element a polynomial of the form $1 + Dp(D)$, where $p(D)$ is a polynomial in D . Since this argument is valid for all pairs $0 \leq i, j \leq m-1, i \neq j$, it follows that all but at most one edge $e_i, 0 \leq i \leq m-1$ must have a first element of the form $1 + Dp(D)$. Without loss of generality, we assume the first element of $\mathbf{v}(e_i)$ is in the form $1 + Dp(D)$ for all $0 \leq i \leq m-2$.

Consider now the second element of the vectors $\mathbf{v}(e_i), \mathbf{v}(e_j)$ such that $0 \leq i, j \leq m-1, i \neq j$. Suppose that the smallest degree additive term of the second element of vectors $\mathbf{v}(e_i), \mathbf{v}(e_j)$ is, respectively D^{q_i} and D^{q_j} . We

begin by demonstrating that if $i \neq j$ and $i, j \neq m-1$, then $q_i \neq q_j$. The proof is by contradiction. If $i, j \neq m-1$, then the first element of $\mathbf{v}(e_i), \mathbf{v}(e_j)$ is in the form $1 + Dp_i(D), 1 + Dp_j(D)$. Assume for now that $q_i = q_j < \infty$. It follows from above that at time $q_i = q_j$ the symbols transmitted on edges e_i, e_j will be, respectively, in the form $x_1(q_i) + f_i(x_1(0), \dots, x_1(q_{i-1})) + x_2(0)$ and $x_1(q_i) + f_j(x_1(0), \dots, x_1(q_{i-1})) + x_2(0)$, where $f_i(\cdot), f_j(\cdot)$ are linear functions of their arguments. Since the deadline is $d_1 = 0$, by the time $n = q_i$ all the symbols $x_1(0), \dots, x_1(q_{i-1})$ will already be decoded at the sink. Therefore, without loss of generality, they can be canceled from the received symbols and it can be assumed that the symbols received at time $n = q_i$ on edges e_i, e_j will both be given by $x_1(q_i) + x_2(0)$. Since the deadline is $d_1 = 0$, it is required to decode $x_1(q_i)$ at time $n = q_i$. But since the symbols on edges e_i, e_j will both be given by $x_1(q_i) + x_2(0)$, it will be impossible to decode $x_1(q_i)$. This contradicts the deadline requirement for $x_1(n)$. In the case $q_i = q_j = \infty$, the sink connected to e_i, e_j will never be able to decode stream $x_2(n)$ since the symbols it receives are independent of the stream $x_2(n)$. Therefore, d_2 will be worse than that of the code in (13).

It follows that for $0 \leq i, j \leq m-2, i \neq j, q_i \neq q_j$. Without loss of generality assume that for any $0 \leq i < j \leq m-2$ we have $q_i < q_j$. The symbol $x_2(0)$ is first used for coding at time q_0 (or possibly earlier by the edge e_{m-1}). Since q_i is a nonnegative integer it follows that for $0 \leq i \leq m-2, q_i \geq i + q_0$. The symbols transmitted on edges e_{m-3}, e_{m-2} prior to time $q_{m-3} \geq m-3 + q_0$ do not depend on the stream $x_2(n)$. The symbol $x_2(0)$ cannot be decoded by the sink connected to the edges e_{m-3}, e_{m-2} prior to time q_{m-3} . Therefore, the delay for the sink connected to e_{m-3}, e_{m-2} for decoding the stream $x_2(n)$ is at least $q_{m-3} \geq m-3$. Thus $m-3$ is a lower bound for the achievable deadline d_2 for $x_2(n)$. For the code in Figure 1, it can be verified that all sinks can decode $x_1(0)$ at time $n = 0$. The value of the power of the least degree additive term of $\det A(D)$ is at most $m-3$ for all sinks. Therefore, the decoding delay is at most $m-3$. The theorem follows. ■

B. General Design Criterion

The example in the last section indicates a design criterion for a code for two data types. The code is designed for the sequential decoder introduced in [9]. Consider the “decoding relation” proved in [9]:

$$D^N X_i(D) = (Y_{e_1}(D)J_{l_{i,1}}(D) + \dots + Y_{e_h}(D)J_{l_{i,h}}(D)) \bmod(D^{N+1}) \quad (14)$$

where $J_{l_{i,j}}(D)$ is element (i, j) of the adjoint matrix of $A_l(D)$ and N is the power of the least degree additive term of $\det A_l(D)$. The edges e_1, \dots, e_h are the edges incoming into sink t_l . The operation $\bmod(D^{N+1})$ means that we take $\bmod(D^{N+1})$ of both sides of the equation. The decoding delay of the sequential decoder is at most N [9].

The LHS of (14) is the first bit of $X_i(D)$ and since the result is given by the RHS term, which we will shortly show how to compute, the first bit of each stream can be decoded.

The computation of $\det(A_l(D))$ is required in order to find the value of N and has to be performed once. Since we are interested in $J_l(D) \bmod(D^{N+1})$ we look only at the suffix of length $N+1$ of each element of $J_l(D)$. From the received signal $Y_{e_i}(D)$ we also need only the symbols that have arrived from instant 0 to instant N . Therefore, the decoding delay for the first symbol is indeed at most N .

Theorem 5: The achievable deadlines for the case of two data types are given by

- 1) Deadline $d_1 = N - \min\{q_1, \dots, q_h\}$ can be achieved for the more time sensitive stream $x_1(n)$, where the power of the smallest degree additive term of $J_{l_{1,i}}(D), 1 \leq i \leq h$ is denoted by q_i .
- 2) Deadline $d_2 = N$ can be achieved for the less time sensitive stream $x_2(n)$.

Proof: Since according to [9] the delay of the sequential decoder for decoding both $x_1(n)$ and $x_2(n)$ is at most N , clearly N is an achievable deadline for $x_2(n)$.

The relation (14) for $i = 1$ is given by

$$D^N X_1(D) = (Y_{e_1}(D)J_{l_{1,1}}(D) + \dots + Y_{e_h}(D)J_{l_{1,h}}(D)) \bmod(D^{N+1}). \quad (15)$$

For the case of two data types, the goal is to minimize both the deadline d_1 of the more time-sensitive data type $x_1(n)$, and the deadline d_2 of the less time-sensitive data type $x_2(n)$. The deadline of $x_2(n)$ is determined by N , since N is the decoding delay before both data types can be decoded. But $x_1(n)$ may have a smaller delay. The term $J_{l_{1,i}}(D)$ can be written as $D^{q_i}p_i(D)$ where $p_i(D)$ is a polynomial in D . Rewriting (15) yields

$$D^N X_1(D) = (Y_{e_1}(D)D^{q_1}p_1(D) + \dots + Y_{e_h}(D)D^{q_h}p_h(D)) \bmod(D^{N+1}). \quad (16)$$

In order for (16) to be maintained for any source stream $x_1(n)$ it follows that $\min\{q_1, \dots, q_h\} \leq N$. Dividing (16) by $D^{\min\{q_1, \dots, q_h\}}$ yields

$$D^{N-\min\{q_1, \dots, q_h\}} X_1(D) = (Y_{e_j}(D)D^{\min\{0, q_1 - \min\{q_2, \dots, q_h\}\}}p_1(D) + \dots + Y_{e_h}(D)D^{\min\{0, q_h - \min\{q_1, \dots, q_{h-1}\}\}}p_h(D)) \bmod(D^{N+1-\min\{q_1, \dots, q_h\}}). \quad (17)$$

From (17) we observe that the LHS is a delayed version of $x_1(n)$ by $N - \min\{q_1, \dots, q_h\}$ (instead of N in (16)). In order to compute the LHS it is required in the RHS for the source to receive the symbols up to time $N - \min\{q_1, \dots, q_h\}$. It follows that a deadline of $N - \min\{q_1, \dots, q_h\}$ can be achieved for $x_1(n)$. ■

Thus, the design criterion to minimize the deadline for $x_1(n)$ is to minimize the values of $N - \min\{q_1, \dots, q_h\}$ while the design criterion in order to minimize the deadline for $x_2(n)$ is to minimize the values of N .

In the example in the last section $h = 2$ and the deadline for $x_1(n)$ was required to be 0. It can be verified that for all sinks $N - \min\{q_1, q_2\}$ is 0. For example, for the sink

connected to e_{m-4} and e_{m-3} , $\det A(D) = D^{m-4} + D^{m-3}$ and therefore $N = m - 4$. For this sink $q_1 = m - 4$ and $q_2 = m - 3$ and so, $N - \min\{q_1, q_2\} = 0$. It follows that the stream $x_1(n)$ maintains deadline equal to 0.

VI. REPRESENTATION OF THE NETWORK AS NON-MULTICAST WITHOUT DEADLINES

An alternative approach for tackling the problem of deadlines is representing multicast with deadlines by a non-multicast problem without deadlines. In the following we describe an approach for constructing network codes with deadlines on an arbitrary graph G . As in [2], we assume without loss of generality that sink nodes have no outgoing edges. Each sink is required to receive by time t all the symbols whose deadline has expired by that time. In this model, each edge might introduce delay. This motivates us to unfold the network into an extended network. This approach is similar to the time slotted network in [1], [2].

Definition 8: Given a network $G = (V, E)$ with a source node $s \in V$, and a positive integer T , the associated time slotted network G^T includes all nodes of the type x^t where $x \in V$ and t ranges through integers 1 and T . For $1 \leq t \leq T$, s^t is the source of bits that can be first used at time t . The edges in G^T are of the following two types:

- For $t \leq T - 1$ and for each node $x \in V$, there is an infinite capacity edge from x^t to x^{t+1} in G^T
- For $t \leq T$ and for each edge $(x, y) \in E$ of delay δ , there is an equal capacity edge from x^t to $y^{t+\delta}$.

The parameter T is the transmission length. Source s^t multicasts information with deadline Δ to the sinks $x^{t+\Delta}$. The resulting problem with T sources is a scalar non-multicast problem without deadlines. Since bits are the most basic unit and each layer in G^T is a single time step, the coding coefficients are binary.

For the case where the source rate R and deadlines Δ are fixed and time-invariant, we show that for linear convolutional codes the complexity of checking the feasibility of the problem and constructing a code, if one exists, can be upper bounded as follows. We use an inductive approach. Assume that sinks with deadlines less than $t + \Delta$ are able to decode successfully. We want to find whether source s^t can transmit successfully to sinks $x^{t+\Delta}$. Consider a reduced network consisting only of those nodes and links that are both downstream of s^t and upstream of sinks $x^{t+\Delta}$, since links that are not downstream of s^t carry only information that is by the induction hypothesis available to sinks $x^{t+\Delta}$ and can be decoded out. The nodes $x^\tau, t \leq \tau \leq t + \Delta$, corresponding to each sink x are coalesced into a single node x . The edge $(x^\tau, x^{\tau+1}), t \leq \tau \leq t + \Delta - 1$, for each non-sink node x is reduced in capacity to $R(\tau - t + 1)$ corresponding to the rate of its upstream sources, and we can set it to forward any full rank transformation of its incoming information. We need to choose binary coding coefficients for each edge of the original graph G , which are then repeated at each time step. Since each edge in G^T has at most $I + R\Delta$ incoming edges where I is the maximum in-degree of a node in G , the number of coding coefficients we need to choose is upper

bounded by $(I + R\Delta)|E|$. A solution can be obtained using the Gröbner bases approach of [12] for general scalar non-multicast problems.

VII. CONCLUSION

In this work we have shown that scalar algebraic codes, which achieve optimal rates for networks without deadlines, are not necessarily optimal for network with deadlines. Furthermore, we have shown that for networks with deadlines, block codes with finite blocklength are also not optimal in general. To show these results, we have analyzed an example network. For that network, it turns out that linear time-invariant convolutional codes are optimal. It would be interesting to prove or disprove that linear time-invariant convolutional codes are optimal in general for network with deadlines. For the case of multiple types of data, where each type of data has a distinct deadline, we have found a design criterion for the code. It would be interesting to find an explicit and efficient code construction for this case.

As an alternative approach, we have shown that the problem of multicast with deadlines can be represented as a non-multicast problem without deadlines. Using that approach, we have found an upper bound on the complexity of checking the feasibility of the problem. It is an interesting question as to whether the special structure of the problem admits lower-complexity solutions than the one that we have found.

REFERENCES

- [1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R.W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.
- [2] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, February 2003.
- [3] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, June 2005.
- [4] M. Feder, D. Ron, and A. Tavor, "Bounds on linear codes for network multicast," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 10, no. 33, 2003.
- [5] A. Rasala-Lehman and E. Lehman, "Complexity classification of network information flow problems," *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 142–150, New Orleans, Louisiana, January 2004.
- [6] C. Fragouli, E. Soljanin, and A. Shokrollahi, "Network coding as a coloring problem," *2004 Conference on Information Sciences and Systems*, Princeton, NJ, March 2004.
- [7] C. K. Ngai and R. W. Yeung, "Network coding gain of combination networks," *IEEE Information Theory Workshop*, pp. 283 – 287, 24–29 Oct. 2004.
- [8] E. Erez and M. Feder, "Convolutional network codes," *IEEE International Symposium on Information Theory*, p. 146, Chicago, IL, June 2004.
- [9] E. Erez, *Topics in Network Coding*, Ph.D. thesis, Tel Aviv University, 2007.
- [10] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, "Network coding theory," *Foundation and Trends in Communications and Information Theory*, vol. 2, no. 4 and 5, pp. 241–381, 2005.
- [11] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, Amsterdam, Netherlands: North Holland Mathematical Library, 1977.
- [12] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, October 2003.