Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2008, Article ID 329727, 7 pages
doi:10.1155/2008/329727

Research Article

Average Throughput with Linear Network Coding over Finite Fields: The Combination Network Case

Ali Al-Bashabsheh and Abbas Yongacoglu

School of Information Technology and Engineering, University of Ottawa, Ottawa, Canada K1N 6N5

Correspondence should be addressed to Ali Al-Bashabsheh, aalba059@site.uottawa.ca

Received 4 November 2007; Revised 17 March 2008; Accepted 27 March 2008

Recommended by Andrej Stefanov

We characterize the average linear network coding throughput, T_c^{avg}, for the combination network with min-cut 2 over an arbitrary finite field. We also provide a network code, completely specified by the field size, achieving T_c^{avg} for the combination network.

1. INTRODUCTION

For a set of sinks in a directed multicast network, it was shown in [1] that if the network can achieve a certain throughput to each sink individually, then it can achieve the same throughput to all sinks simultaneously by allowing coding at intermediate nodes. Such argument is possible since information is an abstract entity rather than a physical one. Thus, in addition to repetition and forwarding, nodes can manipulate the symbols available from their in-edges and apply functions of such symbols to their out-edges. The collection of edge functions can be referred to as the *network code*. The work in [1] shows that a natural bound on the achievable coding rate is the least of the min-cuts between the source and each of the sinks. We refer to a code achieving the min-cut rate as a *solution*.

It is known that every multicast network has a *linear* solution over a sufficiently large finite field [2]. Sanders et al. [3] showed for linear network coding that an alphabet of size $\mathcal{O}(|\mathcal{T}|)$ is sufficient to achieve the min-cut rate, where $|\mathcal{T}|$ is the number of sinks in the network. Rasala-Lehman and Lehman [4] indicated that such bound is tight by advising a solvable multicast network requiring an alphabet of size $\Omega(\sqrt{|\mathcal{T}|})$ to achieve the min-cut rate. This shows that some multicast networks might require alphabets of huge sizes to achieve their min-cut throughputs. Coding over large alphabet sizes is not always desirable since it introduces some complexity and latency concerns. Hence, this motivates working below min-cut rates (i.e., relaxing the constraint of operating at network capacity) but with significantly smaller alphabet sizes (if possible) [5].

Chekuri et al. [6] introduced the measure *average routing throughput* by relaxing the constraint that all sinks must receive the same rate. By decoupling the problem of maximizing the average rate from the problem of balancing rates toward sinks, they showed that average routing rates can significantly exceed the maximum achievable common routing rates in the network. They also argued that the majority rather than the minority of multicast applications experience different rates at the receivers. In [7], the concept *average linear network coding throughput* was introduced under the constraint where source alphabet and linear network coding are restricted to the binary field. In this work, we extend average coding throughput measure to include linear coding over arbitrary finite fields. Such extension is an important step toward practical network coding. To see this, we first remark that in [7] the motivation to restrict the alphabet to the binary field was to present a simple coding scheme where nodes are not required to perform operations over a field larger than $\mathbb{F}_2$. Such cut on processing complexity came with the price of reducing the total network throughput. In practice, although nodes might not posses the capability to perform operations over the necessary field to achieve the min-cut throughput, they might still have a computation capability beyond the binary field. In such situations, it is more reasonable to design codes compatible with nodes computation capability and thus get closer to the min-cut throughput.

In the literature, two different variations of the problem of nonuniform coding throughputs at the terminals were considered. The *general connection model* [5, 8] considers the problem where each sink specifies its set of demanded messages. On the other hand, the *nonuniform demand* problem refers to the problem where each sink specifies only the size of its demand (i.e., the number of messages) and such demanded size might vary from sink to another [9]. In this work, a sink does not specify neither the identities nor the number of demanded messages. The objective is to maximize the average throughput achievable with linear network coding under the additional constraint where messages and network coding are restricted to the finite field $\mathbb{F}_q$ (where $\mathbb{F}_q$ might not be sufficiently large to achieve the min-cut throughput).

2. DEFINITIONS AND PROBLEM FORMULATION

In general, assume an h unit rate information source consists of h unit rate messages $x_1, x_2, \ldots, x_h$, where messages are symbols from a finite field $\mathbb{F}_q$. Also assume symbols carried by edges belong to the same field $\mathbb{F}_q$. For the comparison between average throughputs and common throughputs to be fair and meaningful, it is important that the number of messages h at the source does not exceed the min-cut. In the more general case where the min-cuts from the source to the sinks are not equal, h can be set such that it does not exceed the smallest of such min-cuts.

A directed network $\mathcal{N}$ on V nodes and E links can be modeled as a directed graph $G(V, E)$. In multicast networks, a node $s \in V$ broadcasts a set of messages $x_1, \ldots, x_h$ to a set of sinks $\mathcal{T} = \{t_1, \ldots, t_n\} \subseteq V \setminus s$ where h is the smallest min-cut between s and t, for all $t \in \mathcal{T}$. For any edge $e \in E$, we denote by $\delta_{\text{in}}(e)$ the set of edges entering the node from which e departs. At some parts of this work, we find it more convenient to deal with the *valuation* (defined below) induced by the network code rather than the network code itself.

Definition 1. Given a network code, C, defined by a collection of functions f_e, for all $e \in E$ such that

$$f_e : \mathbb{F}_q^{|\delta_{\text{in}}(e)|} \longrightarrow \mathbb{F}_q. \tag{1}$$

The code *valuation* induced by C is a collection of functions:

$$f_e' : \mathbb{F}_q^h \longrightarrow \mathbb{F}_q, \tag{2}$$

where f_e' is the value of f_e as a function of $x_1, \ldots, x_h$.

For a multicast network $\mathcal{N}$ with a set of messages at the source whose size does not exceed the smallest of the min-cuts between the source and each sink. Linear network coding over sufficiently large field allows every sink $t \in \mathcal{T}$ to recover the entire set of messages. In this work, we somehow reverse the story, that is, we restrict the field size and allow sinks to recover subsets of the set of messages. Since sinks do not experience the same throughput any more, average throughput per sink becomes a natural measure to evaluate the performance of a given network code. Hence, the objective is to decide on a linear network code over the specified alphabet which maximizes the average throughput- or equivalently the sum of throughputs experienced by all sinks. More formally, we define the maximum average linear coding throughput over $\mathbb{F}_q$ as

$$T_c^{\text{avg}} = \frac{1}{|\mathcal{T}|} \max_{Q \in \mathcal{Q}} \left[\sum_{t \in \mathcal{T}} T_c^t(Q) \right], \tag{3}$$

where the maximization is over $\mathcal{Q}$, the set of all possible linear coding schemes over $\mathbb{F}_q$, and $T_c^t(Q)$ is the throughput at sink t under linear coding scheme $Q \in \mathcal{Q}$. In contrast, maximum average routing throughput was defined in [6] and is repeated here for convenience:

$$T_i^{\text{avg}} = \frac{1}{|\mathcal{T}|} \max_{P \in \mathcal{P}} \left[\sum_{t \in \mathcal{T}} T_i^t(P) \right], \tag{4}$$

where in this formulation, the maximization is over all possible integer routing schemes, $\mathcal{P}$, and $T_i^t(P)$ is the integer routing throughput at sink t under routing scheme $P \in \mathcal{P}$.

In what follows, we restrict our attention to the family of combination networks with N intermediate nodes and min-cut 2. Such networks are sufficient to develop the ideas we need to present in this work.

3. COMPLEXITY VERSUS ALPHABET SIZE

Consider a multicast network that requires a field $\mathbb{F}_q$ of size q to achieve the min-cut throughput. Thus, all operations to compute edge functions must be done over the field $\mathbb{F}_q$. In other words, each node in the network must have a memory of $\Theta(\log_2(q))$ bits to store and manipulate the received symbols. In practice, each edge can deliver a fixed number of bits per unit time. Hence, the assumption that edges can deliver one symbol from $\mathbb{F}_q$ per network use implies a latency of $\Theta(\log_2(q))$.

4. AVERAGE THROUGHPUT OF COMBINATION NETWORK WITH MIN-CUT 2

Consider a combination network $\mathcal{N}$ with min-cut 2 and messages $x_1, x_2 \in \mathbb{F}_q$ at the source. A combination network with min-cut 2 consists of three layers of nodes: the source s, a set of N intermediate nodes, and a set of $\binom{N}{2}$ sinks. The source has an out-edge to each intermediate node. Finally, each distinct pair of intermediate nodes is connected to a unique sink via a pair of edges directed into the sink. In this section, we derive an expression for the maximum average throughput of $\mathcal{N}$ over an arbitrary finite field $\mathbb{F}_q$. It is known that the network, $\mathcal{N}$, is solvable when $N \leq q + 1$, where q is the field size. Hence, average throughput is equivalent to the min-cut throughput in this case. On the other hand, for $q = 2$, the problem was solved in [7]. Therefore, in most of the mathematical treatment which follows, we assume $2 < q < N - 1$. In spite of this assumption, whenever applicable, we use the previously obtained results for the binary field and the fact that $\mathcal{N}$ is solvable for $q \geq N - 1$ to present the results for any $q \geq 2$.

Definition 2. Let $\mathcal{F}$ be a collection of functions such that

$$f : \mathbb{F}_q \times \mathbb{F}_q \longrightarrow \mathbb{F}_q, \tag{5}$$

for each $f \in \mathcal{F}$. Then, an average throughput, T_c^{avg}, is said to be achievable over $\mathcal{F}$ if there exists a network code valuation $C = \{f_e(x_1, x_2) : f_e(x_1, x_2) \in \mathcal{F} \text{ for all } e \in E\}$ achieving T_c^{avg}.

Let $\mathcal{G} = \{\alpha x_1 + \beta x_2 : \alpha, \beta \in \mathbb{F}_q\}$ be the set of all linear functions (combinations) of x_1 and x_2 over $\mathbb{F}_q$. Also let $\mathcal{F} = \{x_2\} \cup \{x_1 + \beta x_2 : \beta \in \mathbb{F}_q\}$.

Lemma 1. *Let $f(x_1, x_2) = \alpha x_1 + \beta x_2$, $f'(x_1, x_2) = \alpha' x_1 + \beta' x_2$ be two functions in $\mathcal{G}$ with $\alpha, \beta, \alpha', \beta' \in \mathbb{F}_q \setminus \{0\}$ then x_1 is recoverable from f and f' if and only if x_2 is recoverable from f and f' (i.e., either both messages are recoverable or non of them).*

Proof. See the appendix. □

Corollary 1. *Let $f(x_1, x_2) = x_1 + \beta x_2$, $f'(x_1, x_2) = x_1 + \beta' x_2$ be two functions in $\mathcal{F}$ with $\beta, \beta' \in \mathbb{F}_q \setminus \{0\}$ then x_1 is recoverable from f and f' if and only if x_2 is recoverable from f and f'.*

Proof. The corollary follows from Lemma 1 and the fact that $\mathcal{F} \subset \mathcal{G}$. □

Lemma 2. *An average throughput, T_c^{avg}, is achievable over $\mathcal{F}$ if and only if it is achievable over $\mathcal{G}$.*

Proof. See the appendix. □

Remarks

(i) Lemma 2 suggests that it is sufficient to consider the set of functions $\mathcal{F}$ and there is no gain in considering $\mathcal{G}$.

(ii) With a slight modifications in the proofs, it is easy to show that Lemmas 1 and 2 are still valid even if $\mathcal{G}$ was the set of all affine functions of x_1 and x_2 over $\mathbb{F}_q$.

From the definition of $\mathcal{F}$, we see that $|\mathcal{F}| = q + 1$, and with the aid of Corollary 1, it is straight forward to show that any sink receiving two distinct functions from $\mathcal{F}$ will be able to recover both messages. Thus, for $N \leq q + 1$ the network is solvable [10], and the average throughput is equal to the min-cut throughput. For $N > q + 1$, a simple pigeon hole argument shows that some source edges will be carrying the same combination of messages. Thus, a receiver with two in-edges carrying the same function of messages will be able to recover one or non of the messages.

The next proposition determines how functions $f(x_1, x_2) \in \mathcal{F}$ must be distributed among source out-edges in order to maximize average throughput (i.e., minimize loss in throughput). Let m_i be the number of source edges carrying $f_i(x_1, x_2) = x_1 + \beta_i x_2$, for all $1 \leq i \leq q - 1$, $\beta_i \in \mathbb{F}_q \setminus \{0\}$, $\beta_i \neq \beta_j$, for all $i \neq j$. With such assignment of functions to source out edges, we still have $N - \sum_{i=1}^{q-1} m_i$ unused source out-edges and two functions in $\mathcal{F}$ namely, $f(x_1, x_2) = x_1$ and $f(x_1, x_2) = x_2$. Let m_0 and m_0' be the number of source edges carrying x_1 and x_2, respectively. Since there is no preference in recovering one message over the other (both messages are equally important to each destination), a maximum average throughput achieving assignment must have $N - \sum_{i=1}^{q-1} m_i$ equally divided between m_0 and m_0', that is, $m_0 = \lfloor (N - \sum_{i=1}^{q-1} m_i)/2 \rfloor$ and $m_0' = \lfloor (N - \sum_{i=1}^{q-1} m_i + 1)/2 \rfloor$. Now, if a sink has both its in-edges carrying x_1 then it can not recover x_2, and thus there are $\binom{m_0}{2}$ sinks which cannot recover x_2. Similarly, there are $\binom{m_0'}{2}$ destinations which cannot recover x_1. Finally, a destination receiving $f_i(x_1, x_2) = x_1 + \beta_i x_2$, $\beta_i \neq 0$ on both of its in-edges will not recover any of the messages, and so there is a loss of $2\binom{m_i}{2}$. Hence, the total loss in throughput is given by

$$\begin{aligned} &L(m_1, \ldots, m_k) \\ &= \binom{\left\lfloor \frac{N - \sum_{i=1}^{k} m_i}{2} \right\rfloor}{2} + \binom{\left\lfloor \frac{N - \sum_{i=1}^{k} m_i + 1}{2} \right\rfloor}{2} + 2\sum_{i=1}^{k} \binom{m_i}{2}, \end{aligned} \tag{6}$$

where $k = q - 1$ and the average throughput, as function of $m_1, \ldots, m_k$, is given by

$$T_c^{\text{avg}}(m_1, \ldots, m_k) = \frac{1}{\binom{N}{2}} \left[2\binom{N}{2} - L(m_1, \ldots, m_k) \right]. \tag{7}$$

Before we present the proposition, we need the following lemma.

Lemma 3. *Let*

$$\mathbf{A} = \begin{pmatrix} a & 1 & \cdots & 1 \\ 1 & a & \cdots & 1 \\ \vdots & \cdots & \ddots & 1 \\ 1 & \cdots & 1 & a \end{pmatrix} \tag{8}$$

be a $k \times k$ matrix with $a \in \mathbb{R}$, $a \neq 1$ or $-(k-1)$, then

$$\begin{aligned} \mathbf{A}^{-1} = &\frac{1}{(a-1)(a+k-1)} \\ &\times \begin{pmatrix} a + (k-2) & -1 & \cdots & -1 \\ -1 & a + (k-2) & \cdots & -1 \\ \vdots & \cdots & \ddots & \vdots \\ -1 & \cdots & -1 & a + (k-2) \end{pmatrix}. \end{aligned} \tag{9}$$

Proof. See the appendix. □

Proposition 1. *Average linear network coding throughput of network $\mathcal{N}$ is maximized when $m_1 = m_2 = \cdots = m_k := m_{\text{int}}^*$, where $\lfloor (N+1)/(k+4) \rfloor \leq m_{\text{int}}^* \leq \lfloor (N+1)/(k+4) \rfloor + 1$.*

Proof. From (6), we can write size

$$L(m_1,\ldots,m_k)=\frac{1}{2}\Bigg[\left(\frac{N-\sum_{i=1}^{k}m_i-\delta}{2}\right)\left(\frac{N-\sum_{i=1}^{k}m_i-\delta}{2}-1\right)+\left(\frac{N-\sum_{i=1}^{k}m_i+\delta}{2}\right)\left(\frac{N-\sum_{i=1}^{k}m_i+\delta}{2}-1\right)+4\sum_{i=1}^{k}\frac{m_i(m_i-1)}{2}\Bigg], \tag{10}$$

where $\delta = 0$ and 1 for $N-\sum_{i=1}^{k}m_i$ is even and odd, respectively. This reduces to

$$L(m_1,\ldots,m_k)=\frac{1}{4}\Bigg[\left(N-\sum_{i=1}^{k}m_i\right)^2-2\left(N-\sum_{i=1}^{k}m_i\right)+4\sum_{i=1}^{k}m_i(m_i-1)+\delta\Bigg]. \tag{11}$$

For the moment, we relax the constraintthat $m_1,\ldots,m_k$ must be integer valued. We also relax the constraint that $(N-\sum_{i=1}^{k}m_i)/2$ and $(N-\sum_{i=1}^{k}m_i+1)/2$ must be integers (note that with such relaxation δ disappears from (11)). Now, we compute the partial derivative of (11) with respect to m_j, for all $1\le j\le k$ and equate to zero. Thus, we obtain

$$5m_j+\sum_{i\neq j}m_i=N+1. \tag{12}$$

This can equivalently be written as

$$(m_1\, m_2\,\cdots\, m_k)A=(N+1)(1\,1\,\cdots\,1), \tag{13}$$

where $\mathbf{A}=4\mathbf{I}+\mathbf{1}$, $\mathbf{I}$ is the $k\times k$ identity matrix, and $\mathbf{1}$ is the all ones $k\times k$ matrix. Thus,

$$(m_1\, m_2\,\cdots\, m_k)=(N+1)(1\,1\,\cdots\,1)\mathbf{A}^{-1} \tag{14}$$

and from Lemma 3 (using $a=5$), we obtain

$$(m_1\ m_2\ \cdots\ m_k)=\frac{N+1}{k+4}(1\,1\,\cdots\,1), \tag{15}$$

that is, $m_1=m_2=\cdots=m_k=(N+1)/(k+4)$. Noting that $L(m_1,\ldots,m_k)$ is a convex function of $m_1,\ldots,m_k$ then we know that the integer value m_{int}^{*} of $m_1,\ldots,m_k$ which minimizes $L(m_1,\ldots,m_k)$ is bounded as

$$\left\lfloor\frac{N+1}{k+4}\right\rfloor\le m_{\text{int}}^{*}\le\left\lfloor\frac{N+1}{k+4}\right\rfloor+1 \tag{16}$$

as required. □

Since $T_c^{\text{avg}}=\max_{m_1,\ldots,m_k}T_c^{\text{avg}}(m_1,\ldots,m_k)$ and $T_c^{\text{avg}}(m_1,\ldots,m_k)$ is maximized by the choice of $m_1,\ldots,m_k$ as in Proposition 1, then T_c^{avg} is totally specified by m_{int}^{*}. The following proposition establishes a simple relation between m_{int}^{*} and the field size q.

Proposition 2. *In a combination network with N intermediate nodes, m_{int}^{*} that maximizes the average linear network coding throughput over $\mathbb{F}_q$ is given by*

$$m_{\text{int}}^{*}=\left\lfloor\frac{N+2+\lfloor q/2\rfloor}{q+3}\right\rfloor, \tag{17}$$

where $2<q<N-1$.

Proof. From (7) and (11), and by substituting $m_1=m_2=\cdots=m_k:=m$, we obtain

$$T_c^{\text{avg}}(m)=\frac{1}{2N(N-1)}\big[4N(N-1)-(N-km)^2+2(N-km)-4km(m-1)-\delta\big], \tag{18}$$

where $\delta=0$ for $N-km$ is even, and $\delta=1$ for $N-km$ is odd. Since δ plays a roll in the next derivations, we will write $\delta(N,m)$ to emphasize its dependence on N and m. From (18), we can write

$$T_c^{\text{avg}}(m)=\frac{1}{2N(N-1)}h(m), \tag{19}$$

where

$$h(m)=-(k^2+4k)m^2-2(N+1)km+N(3N-2)-\delta(N,m). \tag{20}$$

Let $m_a=\lfloor(N+1)/(k+4)\rfloor$ and $m_b=m_a+1=\lfloor(N+1)/(k+4)\rfloor+1$, then from Proposition 1 we know that m_{int}^{*} is either m_a or m_b. Thus, we can write

$$m_{\text{int}}^{*}=\arg\max_{m\in\{m_a,m_b\}}T_c^{\text{avg}}(m)=\arg\max_{m\in\{m_a,m_b\}}h(m). \tag{21}$$

In what follows, we assume $k>1$ (the case when $k=1$ was solved in [7])

Now consider the following two possibilities.

Possibility A (k is odd)

Note in this case the field size, q, is even since $k=q-1$. Thus, $q=2^n$ for some integer $n>1$, that is, $\mathbb{F}_q$ is an extension field with characteristic 2. Depending on the parities of N and m_a (or equivalently m_b), the following four cases arise.

Case 1. Both N and m_a are even. Noting that for this case $\delta(N,m_a)=0$ and $\delta(N,m_b)=1$, then from (20) and the fact that $m_b=m_a+1$ we obtain

$$h(m_b)=h(m_a)-k(k+4)(2m_a+1)+2(N+1)k-1. \tag{22}$$

Since $m_a=\lfloor(N+1)/(k+4)\rfloor=(N+1-\epsilon)/(k+4)$ for some $\epsilon\in\{0,1,\ldots,k+3\}$, then we obtain

$$h(m_b)=h(m_a)+k(2\epsilon-(k+4))-1. \tag{23}$$

From this and (21), we see that $m_{\text{int}}^{*}=m_a$ if $k(2\epsilon-(k+4))-1<0$, and $m_{\text{int}}^{*}=m_b$ if $k(2\epsilon-(k+4))-1>0$. But

$k(2\epsilon - (k+4)) - 1 > 0$ if and only if $\epsilon > (k+4)/2 + 1/2k$. Thus,

$$m_{\text{int}}^* = \begin{cases} m_a, & \epsilon < \dfrac{k+4}{2} + \dfrac{1}{2k}, \\ m_b, & \epsilon > \dfrac{k+4}{2} + \dfrac{1}{2k}. \end{cases} \tag{24}$$

Now, we impose more structure on ϵ. Since $m_a = (N+1-\epsilon)/(k+4)$ and noting that m_a is even while $N+1$ and $k+4$ are odd, then ϵ must be odd, that is, $\epsilon \in \{1, 3, \ldots, k, k+2\}$. From this and (24), we get

$$m_{\text{int}}^* = \begin{cases} m_a, & \epsilon \in \left\{1, 3, 5, \ldots, \dfrac{k-1}{2}, \dfrac{k+3}{2}\right\}, \\ m_b, & \epsilon \in \left\{\dfrac{k+7}{2}, \dfrac{k+11}{2}, \ldots, k, k+2\right\}. \end{cases} \tag{25}$$

Case 2. Both N and m_a are odd. An identical result to (25) can be obtained.

Case 3. N is even, and m_a is odd. For this case, we have $\delta(N, m_a) = 1$ and $\delta(N, m_b) = 0$ and from (20) we can write

$$h(m_b) = h(m_a) + k(2\epsilon - (k+4)) + 1, \tag{26}$$

since $m_a = (N+1-\epsilon)/(k+4)$ and from the fact that m_a, $N+1$, and $k+4$ are all odd, then ϵ must be even, that is, $\epsilon \in \{0, 2, \ldots, k+1, k+3\}$.

Now, $k(2\epsilon - (k+4)) + 1 > 0$ if and only if $\epsilon > (k+4)/2 - 1/2k$. Thus,

$$m_{\text{int}}^* = \begin{cases} m_a, & \epsilon \in \left\{0, 2, 4, \ldots, \dfrac{k-3}{2}, \dfrac{k+1}{2}\right\}, \\ m_a, & \epsilon \in \left\{\dfrac{k+5}{2}, \dfrac{k+9}{2}, \ldots, k+1, k+3\right\}. \end{cases} \tag{27}$$

Case 4. N is odd, and m_a is even. An identical result to (27) can be obtained.

Combining the previous four cases, we can write for any odd $k > 1$:

$$m_{\text{int}}^* = \begin{cases} m_a = \left\lfloor \dfrac{N+1}{k+4} \right\rfloor, & \epsilon \in \left\{0, 1, 2, 3, \ldots, \dfrac{k+1}{2}, \dfrac{k+3}{2}\right\}, \\ m_b = \left\lfloor \dfrac{N+1}{k+4} \right\rfloor + 1, & \epsilon \in \left\{\dfrac{k+5}{2}, \dfrac{k+7}{2}, \ldots, k+2, k+3\right\}. \end{cases} \tag{28}$$

Or more compactly

$$m_{\text{int}}^* = \left\lfloor \frac{N + 2 + (k+1)/2}{k+4} \right\rfloor. \tag{29}$$

Possibility B (k is even)

Note that in this case the field size, q, is odd. Thus, $q = p^n$ for some prime $p \neq 2$ and integer $n \geq 1$. As in *possibility A*. the following four cases arise.

Case 1. Both N and m_a are even. In this case, we have $\delta(N, m_a) = \delta(N, m_b) = 0$ and from (20) we can write

$$h(m_b) = h(m_a) + k(2\epsilon - (k+4)), \tag{30}$$

since m_a is even, $N+1$ is odd, and $k+4$ is even, we induce that ϵ must be odd. Combining this with (30), (21) and the fact that $k(2\epsilon - (k+4)) < 0$ if and only if $\epsilon < (k+4)/2$ we obtain, for $k/2$ is even (i.e., k is divisible by 4),

$$m_{\text{int}}^* = \begin{cases} m_a, & \epsilon \in \left\{1, 3, 5, \ldots, \dfrac{k-2}{2}, \dfrac{k+2}{2}\right\}, \\ m_b, & \epsilon \in \left\{\dfrac{k+6}{2}, \dfrac{k+10}{2}, \ldots, k+3\right\}, \end{cases} \tag{31}$$

and for $k/2$ is odd:

$$m_{\text{int}}^* = \begin{cases} m_a, & \epsilon \in \left\{1, 3, 5, \ldots, \dfrac{k}{2}, \dfrac{k+4}{2}\right\}, \\ m_b, & \epsilon \in \left\{\dfrac{k+8}{2}, \dfrac{k+12}{2}, \ldots, k+3\right\}. \end{cases} \tag{32}$$

Case 2. Both N and m_a are odd. Following the same steps as before and noting that ϵ is even for this case, we obtain (for $k/2$ is even)

$$m_{\text{int}}^* = \begin{cases} m_a, & \epsilon \in \left\{0, 2, 4, \ldots, \dfrac{k}{2}, \dfrac{k+4}{2}\right\}, \\ m_b, & \epsilon \in \left\{\dfrac{k+8}{2}, \dfrac{k+12}{2}, \ldots, k, k+2\right\}, \end{cases} \tag{33}$$

and for $k/2$ is odd:

$$m_{\text{int}}^* = \begin{cases} m_a, & \epsilon \in \left\{0, 2, 4, \ldots, \dfrac{k-2}{2}, \dfrac{k+2}{2}\right\}, \\ m_b, & \epsilon \in \left\{\dfrac{k+6}{2}, \dfrac{k+10}{2}, \ldots, k, k+2\right\}. \end{cases} \tag{34}$$

Case 3. N is even, and m_a is odd. It can be shown that m_{int}^* in this case is given by identical relations to (31) and (32).

Case 4. N is odd, and m_a is even. This case can be shown to be similar to Case 2, that is, m_{int}^* is given by (33) and (34).

Combining the previous four cases, we can write for any even $k > 1$:

$$m_{\text{int}}^* = \begin{cases} m_a = \left\lfloor \dfrac{N+1}{k+4} \right\rfloor, & \epsilon \in \left\{0, 1, 2, 3, \ldots, \dfrac{k}{2}, \dfrac{k+2}{2}, \dfrac{k+4}{2}\right\}, \\ m_b = \left\lfloor \dfrac{N+1}{k+4} \right\rfloor + 1, & \epsilon \in \left\{\dfrac{k+6}{2}, \dfrac{k+8}{2}, \ldots, k+2, k+3\right\}. \end{cases} \tag{35}$$

This can also be written as

$$m_{\text{int}}^* = \left\lfloor \frac{N + 2 + k/2}{k+4} \right\rfloor. \tag{36}$$

From (29) and (36), we obtain for any $1 < k < N-2$

$$m_{\text{int}}^* = \left\lfloor \frac{N + 2 + \lfloor (k+1)/2 \rfloor}{k+4} \right\rfloor, \tag{37}$$

substituting $k = q - 1$ and the proposition follows. □

From the work in [7] for $q = 2$ and the results presented in this work, we obtain the following corollary which characterizes T_c^{avg} over any finite field.

Corollary 2. *For a combination network with N intermediate nodes and min-cut 2, the maximum achievable average linear network coding throughput over $\mathbb{F}_q$ is given as*

$$T_c^{\text{avg}} = \begin{cases} 2, & q \geq N-1, \\ \dfrac{1}{\binom{N}{2}}\left[2\binom{N}{2} - L(m_q^*)\right], & 2 \leq q < N-1, \end{cases} \tag{38}$$

where

$$L(m_q^*) = \left(\begin{matrix} \left\lfloor \dfrac{N - km_q^*}{2} \right\rfloor \\ 2 \end{matrix}\right) + \left(\begin{matrix} \left\lfloor \dfrac{N - km_q^* + 1}{2} \right\rfloor \\ 2 \end{matrix}\right) + 2k\binom{m_q^*}{2},$$

$$m_q^* = \begin{cases} \left\lfloor \dfrac{N+2+\lfloor q/2 \rfloor}{q+3} \right\rfloor, & 2 < q < N-1, \\ \left\lfloor \dfrac{N+2}{5} \right\rfloor, & q = 2. \end{cases} \tag{39}$$

Proof. For $q \geq N-1$, the result is immediate since the network is solvable. For $2 < q < N-1$, the claim follows from (6) and (7), where from Proposition 1 we know that average throughput is maximized when $m_1, \ldots, m_k$ are equal. Hence, we substitute m_q^* for $m_1, \ldots, m_k$ in (6), where m_q^* is the integer value which maximizes the average throughput and was obtained in Proposition 2 (it was denoted m_{int}^*). Finally, for $q = 2$ the result was proven in [7]. □

Figure 1 shows T_c^{avg} as a function of N and the number of intermediate nodes, for different field sizes. It also shows the average integer routing throughput T_i^{avg}.

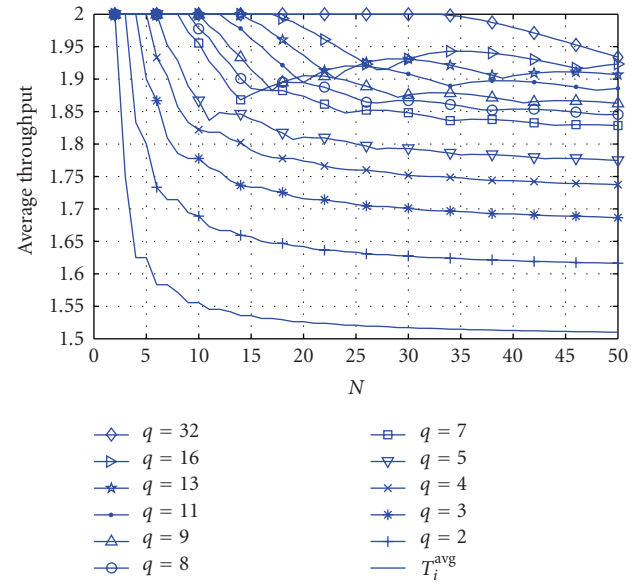

Figure 1: Average linear network coding throughput.

APPENDIX

PROOFS

Proof of Lemma 1. Assume x_1 is recoverable from f and f', then $x_2 = \beta^{-1}(f - \alpha x_1)$ where β^{-1} exists since $\beta \neq 0$, and $\mathbb{F}_q \setminus \{0\}$ is a group under multiplication. Thus, x_2 is recoverable. The proof of the other direction is similar. □

Proof of Lemma 2. The forward implication is obvious since $\mathcal{F} \subset \mathcal{G}$. To prove the reverse implication, assume that there exists a code valuation $\mathcal{C} = \{g_e(x_1, x_2) : g_e(x_1, x_2) \in G, \text{ for all } e \in E\}$ achieving average throughput T_c^{avg}. Consider an indexing set $\mathcal{I} = \{1, 2, \ldots, N\}$ on the source out-edges (and equivalently on intermediate nodes). Since each intermediate node in $\mathcal{N}$ has only one in-edge then intermediate nodes merely forward what they receive on their in-edges to their out-edges. Thus, $\mathcal{C}$ is uniquely specified by the functions carried by the source out-edges. Hence, we may consider $\mathcal{C} = \{g_i(x_1, x_2) : g_i(x_1, x_2) \in \mathcal{G}, \text{ for all } i \in \mathcal{I}\}$. For any $i \in \mathcal{I}$, if $g_i(x_1, x_2) = \alpha_i x_1 + \beta_i x_2 \in \mathcal{C}$ with $\alpha_i = \beta_i = 0$, then g_i can be replaced with any function without reducing the average throughput. Thus, we can assume that α_i and β_i are not both zero. Now, let A be the subset of $\mathcal{I}$ whose elements are the indices of source edges carrying functions $g_i(x_1, x_2) = \alpha_i x_1$, that is, $\beta_i = 0$, for all $i \in A$. Similarly, let B be the subset of $\mathcal{I}$ such that $g_i = \beta_i x_2$, for all $i \in B$. Obviously, A and B are disjoint since α_i and β_i are not both zero for any $i \in \mathcal{I}$. Finally, let C be the subset of $\mathcal{I}$ whose elements are the indices of all source edges carrying functions of the form $g_i = \alpha_i x_1 + \beta_i x_2$ with $\alpha_i \neq 0$ and $\beta_i \neq 0$. Clearly, A, B, and C partition $\mathcal{I}$.

Now, design a code over $\mathcal{F}$ such that $f_i(x_1, x_2) = x_1$, for all $i \in A$, $f_i(x_1, x_2) = x_2$, for all $i \in B$, and $f_i(x_1, x_2) = x_1 + \alpha_i^{-1}\beta_i x_2$, for all $i \in C$. The existence of f_i in the given form for $i \in C$ is guaranteed since $\alpha_i \neq 0$, $\beta_i \neq 0$, and $\mathbb{F}_q \setminus \{0\}$ is a group under multiplication.

Now for all $i, j \in \mathcal{I}$, consider sink $t_{ij} \in \mathcal{T}$ with incoming edges originating from intermediate nodes $i, j \in \mathcal{I}$.

(i) If $i, j \in A$, then t_{ij} will be able to recover only x_1 from g_1 and g_2 which is still the case if g_i and g_j are replaced by f_i and f_j. The same argument holds if $i, j \in B$ with x_2 replacing x_1.

(ii) If $i \in A$ and $j \in B$, then f_i and f_j will make x_1 and x_2 available to t_{ij} so both messages are recoverable and there is no loss in throughput due to replacing g_i and g_j with f_i and f_j.

(iii) If $i \in A$ and $j \in C$, then f_i makes x_1 available to t_{ij} which can be used with f_j to recover x_2. Thus, both messages are recoverable, and there is no loss in considering $\mathcal{F}$ instead of $\mathcal{G}$. A similar argument holds for $i \in B$ and $j \in C$.

(iv) If $i, j \in C$, then from g_i and g_j we can write $\gamma x_2 = \alpha_i g_j - \alpha_j g_i$ where $\gamma = \alpha_i\beta_j - \alpha_j\beta_i$. Hence, x_2 is not recoverable if and only if $\gamma = 0$. From this and Lemma 1 (since $\alpha_i, \beta_i, \alpha_j, \beta_j \in \mathbb{F}_q \setminus \{0\}$), we know that both x_1 and

x_2 are not recoverable if and only if $\gamma = 0$. Also from f_i and f_j, we can write $\gamma' x_2 = f_j - f_i$ where $\gamma' = \alpha_j^{-1}\beta_j - \alpha_i^{-1}\beta_i$, and the same argument for γ holds for γ'. Thus, we need to show $\gamma = 0$ if and only if $\gamma' = 0$. To this end, note that

$$\begin{aligned}\gamma = 0 &\iff \alpha_i\beta_j = \alpha_j\beta_i \iff \beta_j = \alpha_i^{-1}\alpha_j\beta_i \\ &\iff \beta_j = \alpha_j\alpha_i^{-1}\beta_i \iff \alpha_j^{-1}\beta_j = \alpha_i^{-1}\beta_i \iff \gamma' = 0.\end{aligned} \tag{A.1}$$

Hence, there is no loss in considering $\mathcal{F}$ instead of $\mathcal{G}$ in any of the previous cases. The lemma follows by noting that the previous cases represent all possibilities of receiving a pair of functions by any sink. □

Remarks

(i) With a slight modification in the last step of the proof, the lemma can be shown to be still true even if the alphabet was a finite division ring (a skew field) instead of a field.

(ii) It is possible to show that the lemma is true for any multicast network $\mathcal{N}$ with min-cut 2. The proof in this case can be of the same nature as the proof presented in [11] for the sufficiency of homogeneous functions.

Proof of Lemma 3. Note that $\mathbf{A}$ can be written as $\mathbf{A} = (a-1)\mathbf{I} + \mathbf{1}$ where $\mathbf{I}$ is the $k \times k$ identity matrix, and $\mathbf{1}$ is the all ones $k \times k$ matrix ($\mathbf{1}_{ij} = 1$, for all $1 \le i, j \le k$). Let $\mathbf{B}$ be another $k \times k$ matrix such that $\mathbf{AB} = \mathbf{I}$. Assume that $\mathbf{B}$ can be written as $\mathbf{B} = (b\mathbf{I} - \mathbf{1})c$, where b and c are scalars. Thus,

$$\begin{aligned}\mathbf{AB} &= \big((a-1)\mathbf{I} + \mathbf{1}\big)(b\mathbf{I} - \mathbf{1})c \\ &= \big((a-1)b\mathbf{I} + b\mathbf{1} - (a-1)\mathbf{1} - k\mathbf{1}\big)c.\end{aligned} \tag{A.2}$$

For the multiplication $\mathbf{AB}$ to equal $\mathbf{I}$, we need $b\mathbf{1} - (a-1)\mathbf{1} - k\mathbf{1} = \mathbf{0}$, where $\mathbf{0}$ is the all zero matrix. This can be satisfied by choosing

$$b = a + k - 1. \tag{A.3}$$

We also need

$$c(a-1)b = c(a-1)(a+k-1) = 1. \tag{A.4}$$

Since $a \neq 1,\ -(k-1)$, we obtain $c = 1/(a-1)(a+k-1)$. Thus,

$$\mathbf{B} = \frac{1}{(a-1)(a+k-1)}\big((a+k-1)\mathbf{I} - \mathbf{1}\big). \tag{A.5}$$

It is easy to check that $\mathbf{BA} = \mathbf{I}$, thus $\mathbf{A}^{-1} = \mathbf{B}$. □

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, 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, 2003.

[3] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," in *Proceedings of the 15th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '03)*, pp. 286–294, San Diego, Calif, USA, June 2003.

[4] A. Rasala-Lehman and E. Lehman, "Complexity classification of network information flow problems," in *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '04)*, pp. 142–150, New Orleans, La, USA, January 2004.

[5] A. Rasala-Lehman, "Network coding," Ph.D. dissertation, Department of Electrical Engineering and Computer Science, MIT, Cambridge, Mass, USA, 2005.

[6] C. Chekuri, C. Fragouli, and E. Soljanin, "On average throughput and alphabet size in network coding," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2410–2424, 2006.

[7] A. Al-Bashabsheh and A. Yongacoglu, "Average throughput with linear network coding over the binary field," in *Proceedings of the IEEE Information Theory Workshop (ITW '07)*, pp. 90–95, Tahoe City, Calif, USA, September 2007.

[8] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Transactions on Information Theory*, vol. 51, no. 8, pp. 2745–2759, 2005.

[9] Y. Cassuto and J. Bruck, "Network coding for non-uniform demands," in *Proceedings of the International Symposium on Information Theory (ISIT '05)*, pp. 1720–1724, Adelaide, SA, Australia, September 2005.

[10] C. Fragouli and E. Soljanin, "Information flow decomposition for network coding," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 829–848, 2006.

[11] R. Dougherthy, C. Freiling, and K. Zeger, "Linearity and solvability in multicast networks," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2243–2256, 2004.