Electronic Colloquium on Computational Complexity, Report No. 141 (2014)

Linear codes cannot approximate the network capacity within any constant factor

Shachar Lovett *
Computer Science and Engineering
University of California, San Diego
slovett@ucsd.edu

October 24, 2014

Abstract

Network coding studies the capacity of networks to carry information, when internal nodes are allowed to actively encode information. It is known that for multi-cast networks, the network coding capacity can be achieved by linear codes. It is also known not to be true for general networks. The best separation to date is by Dougherty et al. [IEEE Trans. Information Theory, 2005] who constructed a network with capacity 1, where linear codes can achieve a rate of at most 10/11.

We show that linear codes cannot approximate the capacity of a network within any constant factor. That is, for any $0 < \alpha < 1$ we construct a network with network capacity 1, where linear codes can achieve a rate of at most α. This is achieved by a new network composition operation, which allows to amplify bounds on the linear capacity of networks.

1 Introduction

Network information theory studies the capacity of networks to carry information, where internal nodes have the ability to actively encode information. In the network information flow problem, first defined by Ahlswede *et al.* [1], a network is represented by a directed graph containing source nodes which generate messages, sink nodes which demand a subset of the messages and intermediate nodes, which can actively encode information. That is, the information (packets) sent on their out-edges can be an arbitrary function of the packets received on their in-edges. A *network code* is a set of encoding schemes for the intermediate nodes, and decoding schemes for the sink nodes, which allow them to receive all the demanded messages.

A-priori, it is not clear if network coding has any advantage over the classic routing solution, where internal nodes can only route information without changing it. However,

*Supported by NSF CAREER award 1350481.

ISSN 1433-8092

in [1] an example of a network is given (the butterfly network), for which the network coding capacity is twice of that achieved by routing. Specifically, their solution relied on sending linear combinations of the messages over the edges. Subsequently, Li *et al.* [18] showed that linear encodings can attain network capacity in any multi-cast network (a multi-cast network is a network information flow problem, in which all sinks require all messages). Linear solutions for multi-cast networks were further investigated in [3, 4, 12, 14, 15, 17, 21], where in particular it was shown that random linear encodings (over large enough fields) give optimal encoding schemes with high probability.

This suggested the possibility, that perhaps in all network information flow problem, and not just in multi-cast networks, linear encodings always achieve the network capacity [13, 19, 20]. This however turns out to be false. Doughery *et al.* [5] constructed an example of a network information flow problem, for which linear encodings are sub-optimal. Concretely, they constructed a network for which linear encodings are 10/11 worse than the optimal coding solution.

This suggests the following relaxed version of the problem: is it true that linear codes can always approximate the optimal coding solution? This is an interesting question from a number of viewpoints.

- As far as we know, computing the rate of optimal network codes can be undecidable. On the other hand, for linear codes (at least in some regimes) the problem is equivalent to solving polynomial equations [7, 15, 17], which can be solved in exponential time.
- When linear encodings are sub-optimal, it is unknown what is the structure of optimal encodings. In the example of [5], the optimal codes involved linear operations over different finite fields in different intermediate nodes.

Thus, if linear encodings could approximate optimal network codes, we would get at least an approximate answer to both questions. Our main result is that, unfortunately, there are networks in which linear encodings cannot approximate optimal codes within any constant factor. Before explicitly stating the theorem, we first define the computational model formally.

1.1 Formal definitions

Network codes. A network is defined by a directed acyclic multigraph, where we assume information is sent without any delays or errors. Each source node generates a message, which is a sequence of k symbols in a finite alphabet Σ. An edge with capacity n can transmit a packet containing n alphabet symbols. A network code specifies for each internal node, how to compute the packets sent on its out-edges as a function of the packets received in its in-edges, as well as for sink nodes how to decode the required messages from the packets they receive. The coding capacity of a network is the supremum of k/n, over all possible network codings solutions where each message has length k and the maximal edge capacity is n. It is known that the network coding capacity is independent of the alphabet size [2].

Linear network codes. A linear network code is a specific type of a network code. The alphabet is a finite field $\mathbb{F}$. Messages are vectors in $\mathbb{F}^k$, packets are vectors in $\mathbb{F}^n$, and only

$\mathbb{F}$-linear operations are allowed. That is, internal nodes are only allowed to apply linear operations to compute the packets send on their out-edges as a function of the packets received on their in-edges. The linear network coding capacity (over $\mathbb{F}$) is the supremum of k/n, over all possible $\mathbb{F}$-linear network coding solutions. As mentioned before, Doughery *et al.* [5] proved a separation between linear and non-linear codes.

Theorem 1.1 ([5]). *There exists a network with network coding capacity* 1*, but linear network coding capacity at most* $10/11$ *over any finite field.*

The construction of [5] is explicit, and relies on matroids which are non-representable. The proof is also explicit, and involved the study of the system of linear equations arising from these matroids. In this paper, we develop a new approach to prove bounds on the linear capacity of networks, based on a new network composition operation geared towards that goal. Our main result is the following theorem.

Theorem 1.2 (Main Theorem (Theorem 7.2, informal)). *For any* $0 < \alpha < 1$*, there exists a network with network coding capacity* 1*, but linear network coding capacity at most* α *over any finite field. Moreover, the network has size* $\exp(\exp(\log^2 1/\alpha))$ *and depth* $\mathrm{poly}(1/\alpha)$*.*

Sub-constant approximation? As we discussed before, most of the basic questions on the computational model of network codes are unknown. Here, we suggest a couple of new problems, which relate to the ability of linear network codes to at least somewhat approximate the network capacity. The proof of Theorem 1.2 is via a new composition operation for networks, which allows to amplify the gap between the network capacity and linear network capacity. We suspect that the network size can be improved to $\exp(1/\alpha)$ by a more careful amplification process. Beyond that, these bounds seem to be tight in all the examples we are aware of. We propose the following problem, which speculate that for small or low-depth circuits, linear network coding can approximate the network capacity within a small (but sub-constant) factor.

Problem 1.3. Consider a network of size n. Can linear network codes approximate the network capacity up to a factor of $O(\log n)$?

The only known (and trivial) result is than an approximation factor of n is possible. Any improvement over that would be interesting.

Problem 1.4. Consider a network of depth d. Can linear network codes approximate the network capacity up to a factor of $d^{O(1)}$?

In fact, even in constant depth networks, it is unknown whether a constant approximation factor is possible.

1.2 Related works

A number of works showed separation between network capacity and linear network capacity [6, 17, 19, 20], where the best previous separation was by [5]. A somewhat related line of research is on the computational complexity of computing or approximating the capacity or

linear capacity of a network. In [16,22] it is shown that approximating the coding capacity of a network is as hard as approximating the chromatic number of a graph. In [17] it is shown that computing the linear coding capacity of a network is as hard as solving of system of polynomial equations, which is known to be NP-hard. A reduction in the opposite direction, showing that the problems are in fact equivalent, was given in [7].

1.3 Proof overview

We define an operation of network composition between two networks $\mathcal{G}, \mathcal{H}$, resulting in a composed network $\mathcal{K}$, such that two properties hold:

- If $\mathcal{G}, \mathcal{H}$ have network coding capacity 1 then so does $\mathcal{K}$.
- For any finite field $\mathbb{F}$, if $\mathcal{G}$ has linear network coding capacity $r_{\mathcal{G}} < 1$ and $\mathcal{H}$ has a linear network coding capacity $r_{\mathcal{H}} < 1$, then $\mathcal{K}$ has a linear network coding capacity $r_{\mathcal{K}} \leq r_{\mathcal{G}}(1-\delta)$, for some $\delta = \delta(r_{\mathcal{H}}) < 1$.

An iterative application of such an operation implies the required separation. In order to define the composition operation, we focus on unicast networks.

A network is called r-unicast if it has r sources and r sinks, where the i-th sink requires only the message generated by the i-th source, for all $i = 1, \ldots, r$. Any network can be transformed to a unicast network [8], and in particular the example of [5]. Hence, we may choose $\mathcal{H}$ to be a r-unicast network ($r = 10$) with network coding capacity 1 and linear network coding capacity at most 10/11 over any finite field.

Let $\mathcal{G}$ be an arbitrary network. Consider first the following simple composition operation (which is not the composition operation we actually use). Let $\mathcal{G}_1, \ldots, \mathcal{G}_r$ be r disjoint copies of $\mathcal{G}$. For each edge e of $\mathcal{G}$, let $e_1, \ldots, e_r$ be its corresponding copies in $\mathcal{G}_1, \ldots, \mathcal{G}_r$. Now, we "replace" $\{e_1, \ldots, e_r\}$ with a copy of $\mathcal{H}$. That is, if $e_i = (u_i, v_i)$ then we identify $u_1, \ldots, u_r$ with the sources of $\mathcal{H}$, $v_1, \ldots, v_r$ with the sinks of $\mathcal{H}$, remove the edges $e_1, \ldots, e_r$ and add a fresh copy of $\mathcal{H}$. We apply this operation to each edge of $\mathcal{G}$, creating a composed network $\mathcal{K}$.

Now, it is relatively simply to show that if $\mathcal{G}, \mathcal{H}$ have network coding capacity 1 then so does $\mathcal{K}$. This is because each copy of $\mathcal{H}$, corresponding to edges $e_1, \ldots, e_r$, can perfectly simulate the packets sent over the original edges $e_1, \ldots, e_r$, as it has network coding capacity 1. However, what is far less clear is why if $\mathcal{H}$ has linear coding capacity $r_{\mathcal{H}} < 1$, then $\mathcal{K}$ will have a lower linear coding capacity than that of $\mathcal{G}$. The reason is that information can "leak" between different copies of $\mathcal{H}$, as they are all embedded in a single composed graph.

In order to overcome this challenge, we define a more involved composition process, which prevents information from being leaked. Let $\mathcal{G}_1, \ldots, \mathcal{G}_M$ be distinct copies of $\mathcal{G}$ for some large M to be determined later. For each edge e of $\mathcal{G}$, we partition its copies $e_1, \ldots, e_M$ to M/r subsets, and replace each subset with a new copy of $\mathcal{H}$ (we assume that r divides M). The partition of each set of M edges to M/r subsets is defined via a r-uniform hypergraph F defined over M vertices, with edges labeled by edges of $\mathcal{G}$. We show that if the girth of this hypergraph is large enough, then "useful" information cannot leak between copies of $\mathcal{H}$. Finally, we need to show that such hypergraphs exists. It turns out that random hypergraphs of the appropriate parameters do not have large enough girth. However, an adaptation of the Erdös-Sachs theorem [10] allows to construct such hypergraphs when M is large enough.

Paper organization. We review basic definitions in network information theory in Section 2. We define a network composition operation in Section 3. We prove connectivity properties of composed networks in Section 4. We first prove a simple upper bound on the capacity of composed networks in Section 5. We prove an improved upper on the linear capacity of composed networks in Section 6. We combine these to prove our main theorem in Section 7.

2 Network information theory

We recall some basic definitions in network information theory, and refer to [1,11] for details. A *network* $\mathcal{G}$ is a directed acyclic multigraph, with vertex set $V(\mathcal{G})$ and edge set $E(\mathcal{G})$. For an edge $e \in E(\mathcal{G})$ we denote its in- and out-nodes by $in(e), out(e) \in V(\mathcal{G})$, respectively. Some nodes in $V(\mathcal{G})$ are designated as sources or sinks. Each source node is associated with a message, and each sink node demands a subset of the messages generated by the sources. In this paper, we assume all edges have the same capacities, and we restrict our attention to multiple unicast networks. In an r-unicast network, there are r sources $s_1, \ldots, s_r \in V(\mathcal{G})$, r sinks $t_1, \ldots, t_r \in V(\mathcal{G})$, and each sink t_i demands only the message generated by s_i for $i = 1 \ldots r$. We assume without loss of generality that sources are the only nodes with no in-edges, and sinks are the only nodes with no out-edges. Hence, the *depth* of $\mathcal{G}$ is the longest directed path from a source s_i to a sink t_j for $i, j \in [r]$.

Network codes. A *network code* is a recipe for sending information over the edges in the network, in such a way that sinks can reconstruct the required messages from the information they receive. The information carried by an edge e is called a packet, and is given by a function (determined by the network code) of the packets carried by edges whose out-node is the in-node of e. We denote by Σ a finite alphabet, the message generated by a source s is $m(s) \in \Sigma^k$, and each edge e carries a packet $p(e) \in \Sigma^n$. We require that each sink t_i can reconstruct $m(s_i)$ by a function of the packets of its in-edges. In such a case, we say that the network is *solvable* with rate k/n. The *capacity* of the network, denoted $\mathrm{cap}(\mathcal{G})$, is the supremum over all rates with which the network is solvable. It is easy to see that it is invariant to the specific alphabet chosen (see [2] for details).

Linear network codes. A *linear network code* (also called vector-linear) is a special type of a network code. The alphabet is identified with a finite field $\mathbb{F}$, messages lie in the vector space $\mathbb{F}^k$, packets in the vector space $\mathbb{F}^n$, and all functions are restricted to be $\mathbb{F}$-linear maps. When such a linear code exists, we say that the network is *linearly solvable* with rate k/n over the field $\mathbb{F}$. The *linear capacity* (over $\mathbb{F}$) of a network, denoted $\mathrm{cap}_{\mathbb{F}}(\mathcal{G})$, is the supremum of these rates.

Separation between general and linear network codes. Doughery *et al.* [5] constructed a network for which linear network codes achieve a lower rate than the network capacity. This network is not a multiple unicast network. However, in [8] a general method is given to transform any network to a multiple unicast network, which is solvable or linear

solvable over any field if and only if the original network is solvable. A combination of the two results is the base case of our construction.

Theorem 2.1 ([5, 8]). *There exist a* 10*-unicast network* $\mathcal{G}$ *for which* $\text{cap}(\mathcal{G}) = 1$ *and* $\text{cap}_{\mathbb{F}}(\mathcal{G}) \leq 10/11$ *for any finite field* $\mathbb{F}$.

3 Network composition

In this section we define an operation that would allow us to compose two networks which show some separation between linear and general network rates, and obtain a new network with a bigger separation. Let $\mathcal{G}$ be a q-unicast network and let $\mathcal{H}$ be an r-unicast network. Let M be a parameter divisible by r. The starting point is M disjoint copies of $\mathcal{G}$, denoted $\mathcal{G}_1, \ldots, \mathcal{G}_M$. At each iteration, we identify r copies of an edge of $\mathcal{G}$ in different copies, and replace the r parallel edges with a copy of $\mathcal{H}$. Let $v_1, \ldots, v_p$ denote the vertices of $\mathcal{G}$ and $e_1, \ldots, e_m$ denote the edges of $\mathcal{G}$. We denote the vertices of $\mathcal{G}_i$ by $v_{i,1}, \ldots, v_{i,p}$ and the edges of $\mathcal{G}_i$ by $e_{i,1}, \ldots, e_{i,m}$. The specification of which edges to use in each iteration is given by a r-uniform labeled hypergraph.

Definition 3.1 (Matching hypergraph)**.** A *matching hypergraph* for $\mathcal{G}, \mathcal{H}$ is a r-uniform simple hypergraph F on the vertex set $[M]$. That is, the edges of F, denoted $E(F)$, is a family of subsets of $[M]$ of size r. Each edge $f \in E(F)$ is labelled by an index of an edge in $\mathcal{G}$, $L(f) \in [m]$. We require that for every label $\ell \in [m]$, the set of edges

$$\{f \in E(F) : L(f) = \ell\}$$

forms a matching, that is a partition of the vertex set $[M]$.

Definition 3.2 (Network composition)**.** Let $\mathcal{G}$ be a q-unicast network, $\mathcal{H}$ be an r-unicast network, M a parameter diving r, and F a matching hypergraph. The composed network $\mathcal{K} = \text{Compose}(\mathcal{G}, \mathcal{H}, F)$ is defined as follows.

(1) Initialize $\mathcal{K}$ to be $\mathcal{G}_1 \cup \ldots \mathcal{G}_M$, M disjoint copies of $\mathcal{G}$.

(2) For each edge $f \in E(F)$ do the following:

(2.1) Let $i_1, \ldots, i_r$ be an arbitrary ordering of the elements of f. Let $\ell = L(f)$ and assume that $e_\ell = (v_a, v_b)$.

(2.2) Delete from $\mathcal{K}$ the edges $e_{i_j,\ell} = (v_{i_j,a}, v_{i_j,b})$ for $j = 1, \ldots, r$.

(2,3) Let $\mathcal{H}_f$ be a new copy of $\mathcal{H}$. Let $s_{f,1}, \ldots, s_{f,r} \in V(\mathcal{H}_f)$ denote its source nodes and $t_{f,1}, \ldots, t_{f,r} \in V(\mathcal{H}_f)$ denote its sink nodes.

(2,4) For $j = 1, \ldots, r$, add to $\mathcal{K}$ the new edges $e^{in}_{i_j,\ell} = (v_{i_j,a}, s_{f,j})$ and $e^{out}_{i_j,\ell} = (t_{f,j}, v_{i_j,b})$.

We denote edges of the form $e^{in}_{i,\ell}, e^{out}_{i,\ell}$ as "$\mathcal{G}_i$-edges" or sometimes simply as "$\mathcal{G}$-edges". The other edges, belonging to some $\mathcal{H}_f$, are denoted "$\mathcal{H}$-edges". We make a few simple observations: the operations in steps (2.1)-(2.4) operate on distinct edges for different $f \in E(F)$, hence the order in which we enumerate f in step (2) does not matter. The network $\mathcal{K}$ is a (qM)-unicast network.

Next, we note that a network code for $\mathcal{K}$ can be constructed by a composition of network codes $\mathcal{G}$ and for $\mathcal{H}$. This gives a lower bound on the capacity of $\mathcal{K}$.

Lemma 3.3. *Let $\mathcal{G}, \mathcal{H}, \mathcal{K}$ be as defined above. Then* $\text{cap}(\mathcal{K}) \geq \text{cap}(\mathcal{G}) \min(\text{cap}(\mathcal{H}), 1)$ *and* $\text{cap}_{\mathbb{F}}(\mathcal{K}) \geq \text{cap}_{\mathbb{F}}(\mathcal{G}) \min(\text{cap}_{\mathbb{F}}(\mathcal{H}), 1)$ *for any finite field $\mathbb{F}$.*

Proof. We prove the lemma only for general network codes, the case of linear network codes being analogous. Fix an alphabet Σ, a network code for $\mathcal{G}$ with messages of length k and packets of length n, and a network code for $\mathcal{H}$ with messages of length n and packets of length s. By choosing the parameters large enough, we can get k/n and n/s to be arbitrarily close to $\text{cap}(\mathcal{G})$ and $\text{cap}(\mathcal{H})$, respectively. To simplify the presentation we allow packets over edges of $\mathcal{K}$ to have different lengths over different edges, and measure the rate with respect to the largest packet length.

The network code for $\mathcal{K}$ will have the following properties: packets carried along $\mathcal{G}$-edges will have length n and packets carried along $\mathcal{H}$-edges will have length s. Hence, the rate of the network code is $\min(k/n, k/s)$, which by choosing the parameters large enough is arbitrarily close to $\text{cap}(\mathcal{G}) \min(\text{cap}(\mathcal{H}), 1)$.

Our solution will have the property that $p(e^{out}_{i,\ell}) = p(e^{in}_{i,\ell})$ for $i \in [M], \ell \in [m]$ carries a packet which depends only on the messages originating in $\mathcal{G}_i$, and moreover is equal to the packet carried in the network code for $\mathcal{G}$ if the messages were these originating in $\mathcal{G}_i$.

It will be convenient to consider a traversal the edges of $E(\mathcal{G})$ in topological order, although this is not necessary for the formal definition of the code. Consider an edge $e_\ell = (v_a, v_b) \in E(\mathcal{G})$. Assume that in the network code for $\mathcal{G}$, we have that

$$p(e_\ell) = \Gamma_\ell(p(e_{\ell_1}), \ldots, p(e_{\ell_t}), m(v_a))$$

where $e_{\ell_1}, \ldots, e_{\ell_t}$ are the edges for which $out(e_{\ell_1}) = \ldots = out(e_{\ell_t}) = v_a$, and $m(v_a)$ is potential message originating at v_a. We will define the solution over $\mathcal{K}$ for the edges $e^{in}_{i,\ell}$ for $i = 1, \ldots, M$ as

$$p(e^{in}_{i,\ell}) = \Gamma_\ell(p(e^{out}_{i,\ell_1}), \ldots, p(e^{out}_{i,\ell_t}), m(v_{i,a})).$$

Next, consider all edges $f \in E(F)$ for which $L(F) = \ell$. For each such f, apply the network code of $\mathcal{H}$ in the network $\mathcal{H}_f$. That is, if $f = \{i_1, \ldots, i_r\}$, assume the source nodes $s_{f,1}, \ldots, s_{f,r}$ compute "virtual messages" which are the packets sent over their in-edges $e^{in}_{i_1,\ell}, \ldots, e^{in}_{i_r,\ell}$, respectively. By the correctness of the network code in $\mathcal{H}_f$, we get that the sink nodes $t_{f,1}, \ldots, t_{f,r}$ reconstruct these messages, and hence can send these over their respective out-edges $e^{out}_{i_1,\ell}, \ldots, e^{out}_{i_r,\ell}$. That is, we get that $p(e^{out}_{i,\ell}) = p(e^{in}_{i,\ell})$ for all $i \in [M]$.

Finally, the sinks in $\mathcal{G}_1, \ldots, \mathcal{G}_M$ reconstruct their respective required messages from their in-edges in exactly the same way as is done in the network code for $\mathcal{G}$. □

The more challenging aspect is to prove upper bounds on the rates in $\mathcal{K}$, which will show that it is "harder" than $\mathcal{G}$. This will require an additional assumption, that the matching hypergraph does not form short cycles.

4 Connectivity properties of large girth networks

We first formally define the girth of the matching hypergraph F.

Definition 4.1 (Girth of F)**.** A *cycle* of length $g \geq 3$ in F is a sequence of distinct edges $f_1, \ldots, f_g \in E(F)$, such that $f_i \cap f_{i+1} \neq \emptyset$ for all $i = 1, \ldots, g$ where we identify $f_{g+1} = f_1$. The *girth* of F is the length of the shortest cycle in F.

We will show that when girth(F) is large enough, information flow in the network $\mathcal{K}$ is limited. For an edge $e = (u, u') \in E(\mathcal{K})$ define

$$I_e^- = \{i \in [M] : \text{there exists a directed path in } \mathcal{K} \text{ from some } v_i \in V(\mathcal{G}_i) \text{ to } u\}$$

and

$$I_e^+ = \{i \in [M] : \text{there exists a directed path in } \mathcal{K} \text{ from } u' \text{ to some } v_i \in V(\mathcal{G}_i)\}$$

We first show that for $\mathcal{G}_i$-edges form a directed cut between $\{G_{i'} : i' \neq i\}$.

Lemma 4.2. *Assume that* girth$(F) >$ depth(G)*. Let $e \in E(\mathcal{G}_i)$ for some $i \in [M]$. Then $I_e^+ \cap I_e^- = \{i\}$.*

Proof. We prove the lemma for edges of the form $e_{i,\ell}^{in}$. The proof for edges of the form $e_{i,\ell}^{out}$ is analogous. Assume $e_\ell = (v_a, v_b)$ in $\mathcal{G}$ so that $e = e_{i,\ell}^{in} = (v_{i,a}, s_{f,j})$ for some $f \in E(F), j \in [r]$. Clearly $i \in I_e^-$ since $v_{i,a} \in V(\mathcal{G}_i)$. There is a path in $\mathcal{H}_f$ between $s_{f,j}$ to $t_{f,j}$, which is connected by the edge $e_{i,\ell}^{out}$ to $v_{i,b} \in V(\mathcal{G}_i)$. Hence $i \in I_e^+ \cap I_e^-$.

Assume that there exist $i' \in I_e^+ \cap I_e^-$ where $i' \neq i$. That is, there exists a directed path in $\mathcal{K}$ between vertices $v_{i',a'}$ and $v_{i',b'}$ in $V(\mathcal{G}_{i'})$ which passes through e. The $\mathcal{G}$-edges of the path are

$$e_{i_1,\ell_1}^{in}, e_{i_2,\ell_1}^{out}, e_{i_2,\ell_2}^{in}, e_{i_3,\ell_2}^{out}, \ldots, e_{i_d,\ell_d}^{in}, e_{i_{d+1},\ell_d}^{out},$$

where $i_1 = i_{d+1} = i'$, $i \in \{i_2, \ldots, i_d\}$ and $(i_t, i_{t+1}) \in f_t \in E(F)$ for $t = 1, \ldots, d$. Since $L(f_t) = e_{\ell_t}$ and $e_{\ell_1}, \ldots, e_{\ell_d}$ form a directed path in $\mathcal{G}$, we get that $f_1, \ldots, f_d$ are all distinct. Hence, we formed a nontrivial cycle in F of length $d \leq$ depth$(\mathcal{G})$. A contradiction. □

Lemma 4.3. *Assume that* girth$(F) >$ 2depth(G)*. Let $e \in E(\mathcal{G}_i)$, $i^- \in I_e^-$, $i^+ \in I_e^+$, where we exclude the case of $i^- = i^+ = i$. Then any path in $\mathcal{K}$ between a node in $V(\mathcal{G}_{i^-})$ and a node in $V(\mathcal{G}_{i^+})$ must pass through e.*

Proof. Let $e = e_{i,\ell}^*$ where $* \in \{in, out\}$. Assume towards contradiction that there exists a path from some $v_{i^-,a} \in V(\mathcal{G}_{i^-})$ to some $v_{i^+,b} \in V(\mathcal{G}_{i^+})$ which does not pass through e. As we excluded the case $i^- = i^+ = i$, we must have by Lemma 4.2 that $i^- \neq i^+$. The $\mathcal{G}$-edges in the path are

$$e_{i_1,\ell_1}^{in}, e_{i_2,\ell_1}^{out}, e_{i_2,\ell_2}^{in}, e_{i_3,\ell_2}^{out}, \ldots, e_{i_d,\ell_d}^{in}, e_{i_{d+1},\ell_d}^{out},$$

where $i_1 = i^-, i_{d+1} = i^+$ and $(i_t, i_{t+1}) \in f_t \in E(F)$ for $t = 1, \ldots, d$. Note that $e_{\ell_1}, \ldots, e_{\ell_d}$ form a directed path in $\mathcal{G}$, and hence $d \leq$ depth$(\mathcal{G})$ and also $f_1, \ldots, f_d$ are distinct since $L(f_i) = \ell_i$.

Next, since $i^- \in I_e^-, i^+ \in I_e^+$ there exists a path in $\mathcal{K}$ from some vertex $v_{i^-,a'} \in V(\mathcal{G}_{i^-})$ to some $v_{i^+,b'} \in V(\mathcal{G}_{i^+})$ which does pass through e. The $\mathcal{G}$-edges in the path are

$$e^{in}_{i'_1,\ell'_1}, e^{out}_{i'_2,\ell'_1}, e^{in}_{i'_2,\ell'_2}, e^{out}_{i'_3,\ell'_2}, \ldots, e^{in}_{i'_d,\ell'_{d'}}, e^{out}_{i'_{d'+1},\ell'_{d'}},$$

where $i'_1 = i^-, i'_{d'+1} = i^+$ and $(i'_t, i'_{t+1}) \in f'_t \in E(F)$ for $t = 1, \ldots, d'$. Again we have that $d' \leq \text{depth}(\mathcal{G})$ and that $f'_1, \ldots, f'_{d'}$ are distinct.

Let $f^* \in E(F)$ be the unique edge for which $L(f^*) = \ell$ and $i \in f^*$. Then $f^* \notin \{f_1, \ldots, f_d\}$ but $f^* \in \{f'_1, \ldots, f'_{d'}\}$. Let $e \in [d']$ be such that $f'_e = f^*$. Let F' be the hypergraph with $V(F') = V(F) = [M]$ and $E(F') = E(F) \setminus \{f^*\}$. The sequence of edges

$$f'_{e-1}, \ldots, f'_1, f_1, \ldots, f_d, f'_{d'}, \ldots, f'_{e+1}$$

forms a path in F', in the sense that any consecutive pair intersects. Also, $i'_e \in f'_{e-1}, i'_{e+1} \in f'_{e+1}$. This path my contain repeated edges. To fix that, let

$$f''_1, \ldots, f''_{d''}$$

be the shortest sequence of edges in F' such that any consecutive pair intersects and $i'_e \in f''_1, i'_{e+1} \in f''_{d''}$. It is clear that the shortest such sequence would not contain repeated edges. If we concatenate the edge f^* we get a cycle in F of length $d'' + 1 \leq d + d' \leq 2\text{depth}(\mathcal{G})$, a contradiction. □

5 An upper bound on the capacity

We first argue that if F has large girth then the capacity of $\mathcal{K}$ cannot be larger than that of $\mathcal{G}$. That main idea is that we can use a network code for $\mathcal{K}$ also for $\mathcal{G}$, by identifying $\mathcal{G} = \mathcal{G}_1$ and fixing the messages to $\mathcal{G}_2, \ldots, \mathcal{G}_M$.

Lemma 5.1. *Let $\mathcal{G}, \mathcal{H}, \mathcal{K}$ be as defined above. Assume that* $\text{girth}(F) > \text{depth}(\mathcal{G})$. *Then* $\text{cap}(\mathcal{K}) \leq \text{cap}(\mathcal{G})$ *and* $\text{cap}_{\mathbb{F}}(\mathcal{K}) \leq \text{cap}_{\mathbb{F}}(\mathcal{G})$ *for any finite field $\mathbb{F}$.*

Proof. We prove the lemma only for general network codes, the case of linear network codes being analogous. Let $m_1, \ldots, m_q$ denote messages for $\mathcal{G}$. For a network code for $\mathcal{K}$, set $m_1, \ldots, m_q$ to be the messages in $\mathcal{G}_1$ and fix the messages in $\mathcal{G}_2, \ldots, \mathcal{G}_M$ arbitrarily (for linear codes, fix them to 0). Then packets in $\mathcal{K}$ are functions of $m_1, \ldots, m_q$. We will show that for all $\ell \in [m]$, the packet sent over $e^{out}_{1,\ell}$ can be computed as a function of only the packet sent over $e^{in}_{1,\ell}$. This implies that if in $\mathcal{G}$ we set $p(e_\ell) = p(e^{out}_{1,\ell})$, then it satisfies the requirements of a network code. That is, if $e_\ell = (v_a, v_b)$ then $p(e_\ell)$ is a function of the packets of in-edges of v_a as well as $m(v_a)$ when it exists.

Let $f \in E(F)$ be the unique edge such that $L(F) = \ell$ and $1 \in f$. We have that $p(e^{out}_{1,\ell})$ is a function of the inputs to $\mathcal{H}_f$, which are $p(e^{in}_{1,\ell})$ and (potentially some) of $p(e^{in}_{i,\ell})$ for $i \in f \setminus \{1\}$. Denote

$$f' = \{i \in f \setminus \{1\} : \text{there exists a path in } \mathcal{K} \text{ from } e^{in}_{i,\ell} \text{ to } e^{out}_{1,\ell}\}.$$

To conclude the proof, we need to show that for all $i \in f'$, $p(e^{in}_{i,\ell})$ is independent of the messages $m_1, \ldots, m_q$, and hence is fixed since we fixed the messages to $\mathcal{G}_2, \ldots, \mathcal{G}_M$. Assume

not. Then there must exist a path in $\mathcal{K}$ from a source node in $V(\mathcal{G}_1)$ to $v_{i,a}$. However, this implies that $1 \in I^+(e^{in}_{i,\ell}) \cap I^-(e^{in}_{i,\ell})$, which contradicts Lemma 4.2. Hence, $p(e^{in}_{i,\ell})$ for $i \in f'$ becomes fixed once we fix the messages to $\mathcal{G}_2, \ldots, \mathcal{G}_M$, and hence $p(e^{out}_{1,\ell})$ is a function of only $p(e^{in}_{1,\ell})$. □

We obtain so far the following corollary of Lemmas 3.3 and 5.1.

Corollary 5.2. *Let $\mathcal{G}, \mathcal{H}, \mathcal{K}$ be as defined above. Assume that* $\text{girth}(F) > \text{depth}(\mathcal{G})$. *If* $\text{cap}(\mathcal{H}) = 1$ *then* $\text{cap}(\mathcal{K}) = \text{cap}(\mathcal{G})$.

6 An improved upper bound on the linear capacity

We restrict our attention now to linear network codes over a finite field $\mathbb{F}$. We show that if $\text{cap}_{\mathbb{F}}(\mathcal{H}) < 1$ then the "trivial" upper bound $\text{cap}_{\mathbb{F}}(\mathcal{K}) \leq \text{cap}_{\mathbb{F}}(\mathcal{G})$ cannot be achieved. We prove the following theorem in this section.

Theorem 6.1. *Let $\mathcal{G}, \mathcal{H}, \mathcal{K}$ be as defined above, where we assume that* $\text{girth}(F) > 2\text{depth}(\mathcal{G})$ *and* $\text{cap}_{\mathbb{F}}(\mathcal{H}) < 1$. *Then* $\text{cap}_{\mathbb{F}}(\mathcal{K}) \leq (1-\varepsilon)\text{cap}_{\mathbb{F}}(\mathcal{G})$ *where* $\varepsilon = (1 - \text{cap}(\mathcal{H}))/r$.

Fix a linear network code for $\mathcal{K}$. We set a few notations. Sources and sinks of $\mathcal{G}_1, \ldots, \mathcal{G}_M$ are denoted by $s_{i,j}$ and $t_{i,j}$ for $i \in [M], j \in [q]$, with corresponding messages $m_{i,j} = m(s_{i,j}) \in \mathbb{F}^k$. For convenience we set $m(v) = 0$ for any non source node $v \in V(\mathcal{K})$. Recall our convention that the in- and out-nodes of an edge e are denoted by $in(e), out(e)$, respectively. The packet carried over an edge $e \in E(\mathcal{K})$ is denoted $p(e) \in \mathbb{F}^n$, computed as

$$p(e) = S_e m(in(e)) + \sum_{e' \in E(\mathcal{K}): out(e') = in(e)} L_{e,e'} p(e'),$$

such that S_e is an $n \times k$ matrix and $L_{e,e'}$ are $n \times n$ matrices. By linearity, we know that there exist $n \times k$ matrices $P_{e,i,j}$ for $e \in E(\mathcal{K}), i \in [M], j \in [q]$ such that

$$p(e) = \sum_{i=1}^{M} \sum_{j=1}^{q} P_{e,i,j} m_{i,j}.$$

Finally, the requirement that a sink $t_{i,j}$ can reconstruct the message $m_{i,j}$ implies the existence of $k \times n$ matrices $R_{i,j,e}$ such that

$$r_{i,j} = \sum_{e \in E(\mathcal{K}): out(e) = t_{i,j}} R_{i,j,e} p(e).$$

satsifies $r_{i,j} = m_{i,j}$ for all $i \in [M], j \in [q]$.

We first argue that any $\mathcal{G}$-edges can be used to factor the reconstructed messages at the sinks.

Lemma 6.2. *Let $e \in E(\mathcal{G}_{i^*})$ and let $t_{i,j}$ be a sink node with $i^*, i \in [M], j \in [q]$ (note that we allow $i = i^*$). Assume that there exists a path from e to $t_{i,j}$ in $\mathcal{K}$. Define $I \subset [M]$ as*

$$I = ([M] \setminus I^-_e) \cup \{i\}.$$

Then there exist $k \times n$ matrix $A_{i,j,e}$ and $k \times k$ matrices $B_{i,j,i',j'}$, with $i' \in I, j \in [q]$, such that

$$r_{i,j} = A_{i,j,e} p(e) + \sum_{i' \in I, j' \in [q]} B_{i,j,i',j'} m_{i',j'}.$$

Proof. Let $\mathcal{P}_{e',i',j'}$ denote the set of paths from a source $s_{i',j'}$ to an edge $e' \in E(\mathcal{G})$,

$$\mathcal{P}_{e',i',j'} = \{(e_1, \ldots, e_\ell) : e_1, \ldots, e_\ell \in E(\mathcal{K}), in(e_1) = s_{i',j'}, e_\ell = e', \\ out(e_j) = in(e_{j+1}) \; \forall j = 1, \ldots, \ell - 1\}.$$

Then the packet sent over e' is given by $p(e') = \sum_{i' \in [M], j' \in [q]} P_{e',i',j'} m_{i',j'}$ where

$$P_{e',i',j'} = \sum_{(e_1,\ldots,e_\ell) \in \mathcal{P}_{e',i',j'}} L_{e_\ell, e_{\ell-1}} \ldots L_{e_2,e_1} S_{e_1} m_{i',j'}.$$

Similary, let $\mathcal{P}_{i,j,i',j'}$ denote the set of paths from a source $s_{i',j'}$ to a sink $t_{i,j}$,

$$\mathcal{P}_{i,j,i',j'} = \{(e_1, \ldots, e_\ell) : e_1, \ldots, e_\ell \in E(\mathcal{K}), in(e_1) = s_{i',j'}, out(e_\ell) = t_{i,j}, \\ out(e_j) = in(e_{j+1}) \; \forall j = 1, \ldots, \ell - 1\}.$$

Then the reconstructed message at the sink $t_{i,j}$ is given by

$$r_{i,j} = \sum_{i' \in [M], j' \in [q]} \sum_{(e_1,\ldots,e_\ell) \in \mathcal{P}_{i,j,i',j'}} R_{i,j,e_\ell} L_{e_\ell, e_{\ell-1}} \ldots L_{e_2,e_1} S_{e_1} m_{i',j'}.$$

By assumption, $i \in I_e^+$. Hence, by Lemma 4.3 any path from $s_{i',j'}$ to $t_{i,j}$ with $i' \in I_e^-, i' \neq i$ must pass through e. Thus, for any $(e_1, \ldots, e_\ell) \in \mathcal{P}_{i,j,i',j'}$ with $i' \notin I, j' \in [q]$ we have

$$(e_1, \ldots, e_\ell) = (e_1, \ldots, e_{b-1}, e, e_{b+1}, \ldots, e_\ell)$$

for some $b \in [\ell]$, and hence we can factor

$$\sum_{i' \notin I, j' \in [q]} \sum_{(e_1,\ldots,e_\ell) \in \mathcal{P}_{i,j,i',j'}} R_{i,j,e_\ell} L_{e_\ell, e_{\ell-1}} \ldots L_{e_2,e_1} S_{e_1} m_{i',j'} = A_{i,j,e} p(e)$$

where

$$A_{i,j,e} = \sum_{i' \notin I, j' \in [q]} \sum_{(e_1,\ldots,e_{b-1},e,e_{b+1},e_\ell) \in \mathcal{P}_{i,j,i',j'}} R_{i,j,e_\ell} L_{e_\ell, e_{\ell-1}} \ldots L_{e_{b+1},e}.$$

□

We next will apply Lemma 6.2 to show that we can assume, without loss of generality, that the packet carried over each edge $e_{i,\ell}^{out}$ is a linear function of only the packet carried by $e_{i,\ell}^{in}$, and both are linear functions of only messages in $\mathcal{G}_i$.

Lemma 6.3. *There exists a linear network code for $\mathcal{K}$, with messages of length k and packets of size n, in which*

$$p(e_{i,\ell}^{out}) = O_{i,\ell} p(e_{i,\ell}^{in})$$

for any $i \in [M], \ell \in [m]$, where $O_{i,\ell}$ are $n \times n$ matrices. Moreover, both $p(e_{i,\ell}^{in}), p(e_{i,\ell}^{out})$ are linear functions of only $\{m_{i,j} : j \in [q]\}$.

The proof of Lemma 6.3 requires the following simple fact from linear algebra.

Claim 6.4. *Let $V, W \subset \mathbb{F}^n$ be linear subspaces. There exists an $n \times n$ matrix $\Pi = \Pi_{V,W}$ such that*

1. *$\Pi w = 0$ for all $w \in W$.*
2. *$\Pi v \in V$ for all $v \in V$.*
3. *For any $n' \times n$ matrix A which satisfies $Aw = 0$ for all $w \in W$, we have that $A\Pi v = Av$ for all $v \in V$.*

Proof. Fix $x_1, \ldots, x_a, y_1, \ldots, y_b, z_1, \ldots, z_c \in \mathbb{F}^n$ such that $x_1, \ldots, x_a$ is a basis of $V \cap W$, $x_1, \ldots, x_a, y_1, \ldots, y_b$ is a basis for V, and $x_1, \ldots, x_a, y_1, \ldots, y_b, z_1, \ldots, z_c$ is a basis for $\mathbb{F}^n$, where $a + b + c = n$. Define Π as the unique matrix which satisfies $\Pi x_i = 0, \Pi y_j = y_j, \Pi z_k = 0$ for $i \in [a], j \in [b], k \in [c]$. As W is spanned by $x_1, \ldots, x_a, z_1, \ldots, z_c$ we have that $\Pi w = 0$ for all $w \in W$. It is also obvious that $\Pi v \in V$ for all $v \in V$. Finally, let A be a matrix which satisfies $Aw = 0$ for all $w \in W$. Any $v \in V$ can be decomposed as $v = v' + v''$ with v' spanned by $x_1, \ldots, x_a$ and v'' spanned by $y_1, \ldots, y_b$. We have $\Pi v = v''$ and $Av = A(v' + v'') = Av'' = A\Pi v$. □

Proof of Lemma 6.3. Consider a traversal of the edges $e_1, \ldots, e_m$ of $\mathcal{G}$ in a topological order. We will change the packets sent over edges $e^{out}_{i,\ell}$ in this order, each time preserving the network code properties while obtaining that $p(e^{out}_{i,\ell})$ is a linear function of $p(e^{in}_{i,\ell})$, and both are linear functions of $\{m_{i,j} : j \in [q]\}$.

Let $\ell \in [m]$ and consider an edge $e = e^{out}_{i^*,\ell}$ of $\mathcal{K}$. Let $f \in E(f)$ be the unique edge for which $L(f) = \ell$ and $i^* \in f$. By the construction of $\mathcal{K}$, we have that $p(e)$ is a linear function of $\{p(e^{in}_{i,\ell}) : i \in f\}$. In fact, if we let $f_1 \subset f$ be

$$f_1 = \{i \in \text{ there exists a path in } \mathcal{H}_f \text{ from } e^{in}_{i,\ell} \text{ to } e^{out}_{i^*,\ell}\}.$$

then $p(e)$ depends only on $p(e^{in}_{i,\ell})$ for $i \in f_1$. Hence

$$p(e) = \sum_{i \in f_1} Z_i p(e^{in}_{i,\ell}). \tag{1}$$

where Z_i are some $n \times n$ matrices.

By the construction of $\mathcal{K}$, we know that $p(e^{in}_{i,\ell})$ is a linear function of $p(e^{out}_{i,\ell'})$ where $out(e_{\ell'}) = in(e_\ell)$ in $\mathcal{G}$. In particular, $\ell' < \ell$ and hence we already have that $p(e_{i,\ell'})$ is a linear function of $\{m_{i,j} : j \in [q]\}$. So, we have that $p(e^{in}_{i,\ell})$ is also a linear function of $\{m_{i,j} : j \in [q]\}$,

$$p(e^{in}_{i,\ell}) = \sum_{j \in [q]} \eta_{i,\ell,j} m_{i,j} \tag{2}$$

where $\eta_{i,\ell,j} \in \mathbb{F}^n$.

Define subspaces $V, W \subset \mathbb{F}^n$ as

$$V = \text{Span}\{Z_{i^*}\eta_{i^*,\ell,j} : j \in [q]\},$$
$$W = \text{Span}\{Z_i\eta_{i,\ell,j} : i \in f_1 \setminus \{i^*\}, j \in [q]\}.$$

Note that $p(e)$ can obtain any value in $V + W$. Let $\Pi = \Pi_{V,W}$ as defined in Claim 6.4 and define the new packet sent over e to be

$$p_{new}(e) = \Pi p(e).$$

The rest of the network code remains as is. We will show that

(i) $p_{new}(e)$ is a legal packet sent by a linear network code.

(ii) $p_{new}(e)$ is a linear function of only $p(e^{in}_{i^*,\ell})$.

(iii) Each sink $t_{i,j}$ still reconstructs the correct message $m_{i,j}$.

Claim (i) is obvious: if $p(e)$ was a linear function of $\{p(e') : out(e') = in(e)\}$ and $m(in(e))$, then the same holds if we apply any linear map on $p(e)$. Claim (ii) follows immediately from the construction. The main challenge is establishing (iii), that is to show that the change did not damage the abilities of the sinks to reconstruct their respective messages.

Let $t_{i,j}$ be any sink node. If there is no path between e and $t_{i,j}$, then clearly any change to $p(e)$ would not affect the message $r_{i,j}$ reconstructed at $t_{i,j}$. Thus, we assume there exists a path in $\mathcal{K}$ from e to $t_{i,j}$, which implies $i \in I^+_e$. We apply Lemma 6.2 and deduce that

$$r_{i,j} = A_{i,j,e}p(e) + \sum_{i' \in I, j' \in [q]} B_{i,j,i',j'} m_{i',j'}, \tag{3}$$

where $I = (I \setminus I^-_e) \cup \{i\}$, $A_{i,j,e}$ is a $k \times n$ matrix and $B_{i,j,i',j'}$ are $k \times k$ matrices. We need to show that also

$$r_{i,j} = A_{i,j,e}\Pi p(e) + \sum_{i' \in I, j' \in [q]} B_{i,j,i',j'} m_{i',j'}. \tag{4}$$

Consider first the case where $i \neq i^*$. We have that $p(e)$ is a linear function of $\{m_{i',j'} : i' \in f, j' \in [q]\}$. Note that $f \subset I^-_e$ and that $i \notin I^-_e$ since $i \in I^+_e$ and $i \neq i^*$. Hence $f \cap I = \emptyset$ and we get that

$$p(e) = \sum_{i' \notin I, j' \in [q]} P_{e,i',j'} m_{i',j'},$$

where some $P_{e,i',j'}$ can potentially be zero, and

$$r_{i,j} = \sum_{i' \notin I, j' \in [q]} (A_{i,j,e}P_{e,i',j'}) m_{i',j'} + \sum_{i' \in I, j' \in [q]} B_{i,j,i',j'} m_{i',j'}.$$

Since we know that $r_{i,j} = m_{i,j}$ and since $i \in I$, we must have that $A_{i,j,e}P_{e,i',j'} = 0$ for all $i' \notin I, j' \in [q]$, and hence $A_{i,j,e}p(e) = 0$ for any potential messages. Since $p(e)$ can take any value in $V + W$, this implies that $A_{i,j,e}v = A_{i,j,e}w = 0$ for any $v \in V, w \in W$. But since $\Pi p(e) \in V + W$, this implies that also $A_{i,j,e}\Pi p(e) = 0$ for any potential messages. So, we conclude that in the case $i \neq i^*$, we have

$$A_{i,j,e}p_{new}(e) = A_{i,j,e}p(e) = 0. \tag{5}$$

Next, consider the case where $i = i^*$. Combining (3) and (1), we obtain that

$$r_{i^*,j} = A_{i^*,j,e} \sum_{i_1 \in f_1} Z_{i_1} p(e^{in}_{i_1,\ell}) + \sum_{i' \in I, j' \in [q]} B_{i,j,i',j'} m_{i',j'}. \tag{6}$$

We still have $f \subset I^-_e$ with $i^* \in f$, and hence $f \cap I = \{i^*\}$. Similar to the previous argument, since $r_{i^*,j} = m_{i^*,j}$, for any $i_1 \in f_1 \setminus \{i^*\}$ we must have that $A_{i^*,j,e} Z_{i_1} = 0$. Thus, $A_{i^*,j,e} w = 0$ for all $w \in W$. Since $\Pi w = 0$ we also have $A_{i^*,j,e} \Pi w = 0$ for all $w \in W$. By Claim 6.4, we know that for any $v \in V$, and in particular for $v = Z_{i^*} p(e^{in}_{i^*,\ell})$, we have $A_{i^*,j,e} \Pi v = A_{i^*,j,e} v$. Thus, we deduce that also for $i = i^*$ we have

$$A_{i^*,j,e} p_{new}(e) = A_{i^*,j,e} p(e) = A_{i^*,j,e} Z_{i^*} p(e^{in}_{i^*,\ell}). \tag{7}$$

The lemma follows from (5) when $i \neq i^*$ and (7) when $i = i^*$. □

We assume from now on that the linear network code satisfies the conclusion of Lemma 6.3. For an edge $e \in E(\mathcal{G})$ define $\dim(e)$ to be the dimension of the linear subspace spanned by all possible packets that are sent over e. Clearly $\dim(e) \leq n$. We next show that in each $\mathcal{H}_f$, the input and output edges cannot all have dimension very close to n.

Lemma 6.5. *Fix $f = \{i_1, \ldots, i_r\} \in E(F)$ with $L(f) = \ell$. Let $\varepsilon = (1 - \mathrm{cap}(\mathcal{H}))/r$. Then*

$$\sum_{j=1}^{r} \dim(e^{out}_{i_j,\ell}) \leq (1 - \varepsilon) r n.$$

Proof. Consider the restriction of $\mathcal{K}$ to $\mathcal{H}_f$ together with the edges $e^{in}_{i,\ell}, e^{out}_{i,\ell}$ for $i \in f$. Let $p_j = p(e^{in}_{i_j,\ell})$ and $q_j = p(e^{out}_{i_j,\ell})$ be the corresponding packets entering and leaving $\mathcal{H}_f$. As each p_j is a function of distinct inputs (messages in $\mathcal{G}_{i_j}$), there are no correlations between the values that each p_j can take. Formally, if we denote by $V_j \subset \mathbb{F}^n$ the subspace spanned by all potential packets p_j, then $(p_1, \ldots, p_r) \in V_1 \times \ldots \times V_r$ may take all possible values. We further know that $q_j = O_j p_j$ where $O_j = O_{i_j,\ell}$ is some $n \times n$ matrix.

Hence, we can consider the restriction of $\mathcal{K}$ to $\mathcal{H}_f$ as linear network code over $\mathcal{H}$. Each source $s_j = s_{f,i_j}$ is associated with the message $p_j \in V_j$, and each sink $t_j = t_{f,i_j}$ reconstructs $q_j = O_j p_j$. Let $W_j \subset V_j$ be the maximal dimension subspace so that $W_j \cap \ker(O_j) = \{0\}$. Then O_j acts injectively on W_j, $\dim(q_j) = \dim(W_j)$ and there exist a matrix O'_j so that

$$O'_j O_j w_j = w_j \qquad \forall w_j \in W_j.$$

This means that we extracted a network code for $\mathcal{H}$, where packets have length n, and where messages have sizes $k_1 = \dim(W_1), \ldots, k_r = \dim(W_r)$. This is possible only if $\min(k_1, \ldots, k_r) \leq \mathrm{cap}(\mathcal{H}) \cdot n$, which implies that

$$\sum_{j=1}^{r} \dim(e^{out}_{i_j,\ell}) = \sum_{j=1}^{r} \dim(W_j) \leq ((r-1) + \mathrm{cap}(\mathcal{H}))n = (1 - \varepsilon) r n.$$

□

We next show that we can restrict the linear network code of $\mathcal{K}$ to $\mathcal{G}_i$ and obtain a network code for $\mathcal{G}$. For simplicity of exposition, we allow packets to have different lengths over different edges.

Lemma 6.6. *For each $i \in [M]$, there exist a linear network code over $\mathcal{G}$ with messages of length k, and where the packet sent over e_ℓ has length* $\dim(e_{i,\ell})$.

Proof. We may assume, by applying basis changes, that in the linear network code over $\mathcal{K}$ we have $p(e) \in \mathbb{F}^{\dim(e)}$ for all $e \in E(\mathcal{K})$. Consider the following network code over $\mathcal{G}$:

- The message originating at source s_j is $m_j = m_{i,j} \in \mathbb{F}^k$.
- The packet sent over edge e_ℓ is $p(e_\ell) = p(e_{i,\ell}^{out}) \in \mathbb{F}^{\dim(e_{i,\ell}^{out})}$.
- Each sink node t_j reconstructs m_j.

We need to show why this is possible using linear network codes. In the network code over $\mathcal{K}$ we know that each $p(e_{i,\ell}^{out})$ is a linear function of $p(e_{i,\ell}^{in})$, which in turn is a linear function of $\{p(e_{i,\ell'}^{out}) : out(e_{\ell'}) = in(e_\ell)\}$ and potentially $m(in(e_{i,\ell}^{in}))$. Hence, $p(e_\ell)$ is a linear function of only $\{p(e_{\ell'}) : out(e_{\ell'}) = in(e_\ell)\}$ and $m(in(e_\ell))$, which can be realized by a linear code over $\mathcal{G}$. Reconstruction of the messages by sink nodes follows the same logic: the message reconstructed at sink $t_{i,j}$ can be assumed to be a linear function of $\{p(e_{i,\ell}^{out}) : out(\ell) = j\}$, which again is realizable over $\mathcal{G}$. □

We combine Lemma 6.3, Lemma 6.5 and Lemma 6.6 to prove Theorem 6.1.

Proof of Theorem 6.1. Apply Lemma 6.6 to each $i \in [M]$ independently and concatenate the resulting network codes. This achieves a linear network code for $\mathcal{G}$ with messages of length Mk and where the packet sent over an edge e_ℓ has length $\sum_{i=1}^{M} \dim(e_{i,\ell}^{out})$. By Lemma 6.5, we know that for each $f \in E(F)$ for which $L(f) = e$ we have $\sum_{i \in f} \dim(e_{i,\ell}^{out}) \leq (1-\varepsilon)rn$. Since $\{f \in E(f) : L(f) = \ell\}$ partition $[M]$, we obtain that $\sum_{i=1}^{M} \dim(e_{i,\ell}^{out}) \leq (1-\varepsilon)Mn$. Hence

$$\frac{Mk}{(1-\varepsilon)Mn} \leq \mathrm{cap}_{\mathbb{F}}(\mathcal{G})$$

which implies that $k/n \leq (1-\varepsilon)\mathrm{cap}_{\mathbb{F}}(\mathcal{G})$. As $\mathrm{cap}_{\mathbb{F}}(\mathcal{K})$ is the supremum over k/n for all linear network codes for $\mathcal{K}$ over $\mathbb{F}$, we obtain the desired bound

$$\mathrm{cap}_{\mathbb{F}}(\mathcal{K}) \leq (1-\varepsilon)\mathrm{cap}_{\mathbb{F}}(\mathcal{G}).$$

□

7 Separation between capacity and linear capacity

We obtain a separation between capacity and linear capacity by applying network composition iteratively. We first prove that there exists a good choice for the matching hypergraph F. In the following, we have $m = |E(\mathcal{G})|, g = 2\mathrm{depth}(\mathcal{G})$. The proof of the following lemma was suggested to us by Noga Alon.

Lemma 7.1. *Let $r, m, g \geq 1$ be parameters. Let $M = (mr)^{g+4}$. Then there exists a r-uniform hypergraph F on vertex set $[M]$, which is a union of m perfect matchings, such that* $\mathrm{girth}(F) > g$.

Proof. The proof is a variant of the Erdös-Sachs theorem [10] adjusted to hypergraphs. The argument for regular hypergraphs appeared already in [9]. Here, we note that it can be adapted to yield regular hypergraphs which are a union of perfect matchings.

Initialize $F_1, \ldots, F_m$ to be arbitrary perfect matchings and let $F = \cup F_e$ be the resulting multi-hypergraph, which may contain parallel edges. Assume that $\mathrm{girth}(F) = g_0 \leq g$. Let n_0 denote the number of edges in F which participate in a cycle of length g_0. We will define a process that reduces n_0 by at least one. Hence, applying it iteratively will eventually increase the girth of F to more than g. We note that if there are parallel edges then $\mathrm{girth}(F) = 3$, so as soon as $\mathrm{girth}(F) \geq 4$ the matchings $F_1, \ldots, F_m$ will be disjoint.

Let $f_1 \in E(F)$ be an edge which participates in a cycle of length g_0. Assume without loss of generality that $f_1 \in E(F_1)$. We first argue that we can find $f_2, \ldots, f_r \in E(F_1)$ such that $\mathrm{dist}_F(f_i, f_j) > g$, where $\mathrm{dist}_F(f', f'')$ is defined as the smallest $d \geq 0$ for which there exist edges $\tilde{f}_0, \tilde{f}_1, \ldots, \tilde{f}_{d+1} \in E(F)$ such that $\tilde{f}_0 = f', \tilde{f}_{d+1} = f''$ and $\tilde{f}_i \cap \tilde{f}_{i+1} \neq \emptyset$ for all $0 \leq i \leq d$. To see that, note that the number of edges in F whose distance from f_1 is at most g is bounded by

$$|\{f' \in E(f) : \mathrm{dist}_F(f_1, f') \leq g\}| \leq \sum_{i=1}^{g} rm \cdot ((r-1)(m-1))^i \leq (rm)^{g+2}.$$

Hence, since $|E(F_1)| = M/r > (rm)^{g+2}$ there must exist an edge $f_2 \in E(F_1)$ with $\mathrm{dist}(f_1, f_2) > g$. Repeating this for $i = 3, \ldots, r$, where at every time we exclude edges of distance at most g from $f_1, \ldots, f_{i-1}$, we get that as long as $|E(F_1)| = M/r > (i-1)(rm)^{g+2}$ we can find f_i of distance more than g from $f_1, \ldots, f_{i-1}$.

Note that $f_1, \ldots, f_r$ are disjoint, hence $|f_1 \cup \ldots \cup f_r| = r^2$. Let $f'_1, \ldots, f'_r$ be new edges defined over $f_1 \cup \ldots \cup f_r$ such that $|f'_i \cap f_j| = 1$ for all $i, j \in [r]$. Let F' be the hypergraph with $E(F') = E(F) \setminus \{f_1, \ldots, f_r\} \cup \{f'_1, \ldots, f'_r\}$. We claim that no edge in $f'_1, \ldots, f'_r$ participates in a cycle of length g_0. This will show that the number of edges in F' which participate in a cycle of length g_0 must be at most $n_0 - 1$, hence replacing F with F' would certify our assertion.

So, assume without loss of generality, towards contradiction, that f'_1 participates in a cycle of length at most g_0 in F', and let $f''_1, f''_2, \ldots, f''_{g''} \in E(F')$ be that cycle with $f''_1 = f'_1$ and $g'' \leq g_0$. Let $i_1 \in f''_1 \cap f''_2, \ldots, i_{g''} \in f''_{g''} \cap f''_1$ be some elements in intersections of consecutive pairs. Consider first the case that $f''_2, \ldots, f''_{g''} \in E(F) \cap E(F')$. Note that $i_1, i_{g''}$ must belong to different edges in $\{f_1, \ldots, f_r\}$, say $i_1 \in f_{j'}, i_{g''} \in f_{j''}$. This implies that $\mathrm{dist}_F(f_{j'}, f_{j''}) \leq g'' - 1 < g_0$, a contradiction. The other option is that $f''_j \in E(F') \setminus E(F)$ for some $j \neq 1$, and take the minimal such j. We cannot have $j = 2$ or $j = g''$ since $\{f'_1, \ldots, f'_r\}$ are disjoint. So, the sequence $f''_2, \ldots, f''_{j-1}$ forms a path in F with $i_1 \in f''_2, i_{j-1} \in f''_{j-1}$. If i_1, i_{j-1} belong to different edges of $\{f_1, \ldots, f_r\}$ we reach a contradiction to the assumption of the minimal distance. So, we must have that $i_1, i_{j-1} \in f_k$ for some $k \in [r]$. But then $f_k, f''_2, \ldots, f''_{j''-1}$ is a cycle in F of length $j'' < g'' \leq g_0$, a contradiction to the assumption that g_0 is the girth of F. □

We now state our main theorem formally.

Theorem 7.2. *For an infinite sequence of values of N, there exist a multiple unicast network $\mathcal{G}$ on N vertices with $\mathrm{cap}(\mathcal{G}) = 1$ but $\mathrm{cap}_{\mathbb{F}}(\mathcal{G}) \leq \exp(-c\sqrt{\log\log N})$ for any finite field $\mathbb{F}$, where $c > 0$ is an absolute constant.*

Proof. Let $\mathcal{H}$ be the base 10-unicast network given in Theorem 2.1 with a constant number of nodes and edges, and where $\mathrm{cap}(H) = 1$ and $\mathrm{cap}_{\mathbb{F}}(\mathcal{H}) \leq 10/11$ for any finite field. In the following set $\varepsilon = 1/110$. Define a sequence of networks $\mathcal{G}_1, \mathcal{G}_2, \ldots$ as follows. $\mathcal{G}_1 = \mathcal{H}$. To define $\mathcal{G}_{i+1}$, let $m_i = |E(\mathcal{G}_i)|, d_i = \mathrm{depth}(\mathcal{G}_i)$. Apply Lemma 7.1 with $m = m_i, g = 2d_i, r = 10$ to obtain a matching hypergraph F_i for $\mathcal{G}_i, \mathcal{H}$ with $|V(F_i)| = M_i = (10m_i)^{2d_i+4}$. Define $\mathcal{G}_{i+1} = \mathrm{Compose}(\mathcal{G}_i, \mathcal{H}, F_i)$. Theorem 2.1, Corollary 5.2 and Theorem 6.1 imply that

$$\mathrm{cap}(\mathcal{G}_i) = 1, \qquad \mathrm{cap}_{\mathbb{F}}(\mathcal{G}_i) \leq (1-\varepsilon)^i \text{ for any field } \mathbb{F}.$$

Note that the obtained networks have constant maximum degree, hence $|E(\mathcal{G}_i)| = O(V(\mathcal{G}_i))$. Furthermore, $d_{i+1} \leq (\mathrm{depth}(\mathcal{H}) + 1)d_i$ and $m_{i+1} \leq M_i m_i(|E(\mathcal{H}) + r)$. Solving this gives $d_i = \exp(O(i))$ and $\log m_i = O(d_{i-1} \cdot \log m_{i-1}) = \exp(O(i^2))$. So, if we set $N_i = |V(\mathcal{G}_i)|$ we get that $\mathrm{cap}_{\mathbb{F}}(G_i) = \exp(-c\sqrt{\log\log N_i})$ for some absolute constant $c > 0$. □

Acknowledgements. I thank Noga Alon for helpful discussions and suggestions on the Erdös-Sachs theorem and it extensions. I thank Dong Ki Kim for discussions on this and related projects.

References

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 46(4):1204–1216, 2000.

[2] J. Cannons, R. Dougherty, C. Freiling, and K. Zeger. Network routing capacity. *Information Theory, IEEE Transactions on*, 52(3):777–788, 2006.

[3] P. A. Chou, Y. Wu, and K. Jain. Practical network coding. In *Proceedings of the annual Allerton conference on communication control and computing*, volume 41, pages 40–49. The University; 1998, 2003.

[4] R. Dougherty, C. Freiling, and K. Zeger. Linearity and solvability in multicast networks. *Information Theory, IEEE Transactions on*, 50(10):2243–2256, 2004.

[5] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of linear coding in network information flow. *Information Theory, IEEE Transactions on*, 51(8):2745–2759, 2005.

[6] R. Dougherty, C. Freiling, and K. Zeger. Unachievability of network coding capacity. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2365–2372, 2006.

[7] R. Dougherty, C. Freiling, and K. Zeger. Linear network codes and systems of polynomial equations. *Information Theory, IEEE Transactions on*, 54(5):2303–2316, 2008.

[8] R. Dougherty and K. Zeger. Nonreversibility and equivalent constructions of multiple-unicast networks. *Information Theory, IEEE Transactions on*, 52(11):5067–5077, 2006.

[9] D. Ellis and N. Linial. On regular hypergraphs of high girth. *arXiv preprint arXiv:1302.5090*, 2013.

[10] P. Erdös and H. Sachs. Reguläre graphen gegebener taillenweite mit minimaler knotenzahl. *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe*, 12:251–257, 1963.

[11] C. Fragouli, J.-Y. Le Boudec, and J. Widmer. Network coding: an instant primer. *ACM SIGCOMM Computer Communication Review*, 36(1):63–68, 2006.

[12] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong. Toward a random operation of networks. *submitted to IEEE Trans. Inform. Theory*, 2004.

[13] S. Jaggi, M. Effros, T. Ho, and M. Médard. On linear network coding. In *Proc. of the 42nd Allerton Conference*, 2004.

[14] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. Tolhuizen. Polynomial time algorithms for multicast network code construction. *Information Theory, IEEE Transactions on*, 51(6):1973–1982, 2005.

[15] R. Koetter and M. Médard. An algebraic approach to network coding. *Networking, IEEE/ACM Transactions on*, 11(5):782–795, 2003.

[16] M. Langberg and A. Sprintson. On the hardness of approximating the network coding capacity. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 315–319. IEEE, 2008.

[17] A. R. Lehman and E. Lehman. Complexity classification of network information flow problems. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 142–150. Society for Industrial and Applied Mathematics, 2004.

[18] S.-Y. Li, R. W. Yeung, and N. Cai. Linear network coding. *Information Theory, IEEE Transactions on*, 49(2):371–381, 2003.

[19] M. Médard, M. Effros, D. Karger, and T. Ho. On coding for non-multicast networks. In *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, volume 41, pages 21–29. The University; 1998, 2003.

[20] S. Riis. Linear versus non-linear boolean functions in network flow. In *38th Annual Conference on Information Science and Systems (CISS), Princeton, NJ*, 2004.

[21] A. Tavory, M. Feder, and D. Ron. Bounds on linear codes for network multicast. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 10, page 033. Citeseer, 2003.

[22] H. Yao and E. Verbin. Network coding is highly non-approximable. In *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pages 209–213. IEEE, 2009.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de