Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2015-01. Published by the Egerváry Research Group, Pázmány P. sétány 1/C, H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres. ISSN 1587–4451.

Graph independent field size bounds on failure protecting network codes

Erika R. Bérczi-Kovács

January 2015

Graph independent field size bounds on failure protecting network codes

Erika R. Bérczi-Kovács

Abstract

Network coding is an information transmission method with several possible advantages compared to simple routing. Specifically, possible arc failures of a network during a multicast transmission can be tackled by a failure protecting network code, which requires minimal occupation of the network but enables instant recovery. In this paper we show efficient algorithms and lower field size bounds for network code construction that protects any set of arc failures of size at most d. Best known results from Harvey, Karger and Murota [3] gave a polynomial time algorithm for failure protecting network code construction, however, need a field size of at least $|T|\left(\binom{m}{d}+\ldots+\binom{m}{0}\right)$ for the construction, T and m denoting the set of receivers and the number of arcs in the graph, respectively. We give a better lower bound on the required field size that is a function of $|T|, k$ and d, where k denotes the number of messages sent in the network. This new bound is independent of the network size, and also yields a faster algorithm for the problem. The proof is based on results from network encoding complexity and uses similar techniques as Rouayheb, Soljanin and Sprintson in [2].

1 Introduction

In every algorithm for network code construction the required field size is an important parameter, because efficient algorithms require small field sizes. As an example, for the classical multicast linear network code construction in acyclic graphs, Li, Yeung and Cai in [7] showed that the max flow-min cut property is necessary and sufficient for the existence, but their lower bound on a sufficient field size depends on the size of the graph. It was Koetter and Médard in [5] who showed that actually a lower bound of $O(|T|k)$ is enough. With other words, the required field size does not depend on the size of the graph. Later Jaggi et al. [4] improved this lower bound to $|T|$. Our result can be regarded as a step in an analogue series of results for failure protecting network codes. Harvey et al. showed that for the existence of failure protecting network codes a natural min-max type condition is not only necessary but also sufficient [3]. In their approach, the lower bound for the required field size depends on the size of the graph. In this paper we show that a lower bound on the field size can be given which is independent from the number of nodes or arcs in the graph. Our proofs rely on the connection with the topic of network encoding complexity. Ideas used in the proofs are basically similar to the techniques used in [2] for achieving better field size bounds in wiretap networks.

2 Problem formulation

2.1 Network codes from local coefficients

We assume the reader is familiar with the concept of network coding. Here we define notions used for the proof from the viewpoint of local coding coefficients, because they play a central role in failure protecting network codes.

Let $\mathbb{F}_q$ be a finite field of size q and let $\mathbb{F}_q^k$ denote the k-dimensional vector space over $\mathbb{F}_q$, where k is the number of messages to be sent in the network. Let $\mathbf{e}_i$ denote the ith unit vector. For a set $S \subseteq \mathbb{F}_q^k$ of vectors, we denote the linear subspace spanned by S by $\langle S \rangle$.

Definition 2.1. A **network** is an acyclic directed graph $D(V, A)$ with a single **source** node $s \in V$ and a set of **receiver** nodes (assume sinks) $T \subseteq V - s$. Nodes in $V \backslash (T+s)$ are **internal** nodes. Let $L \subseteq A \times A$ be the set of consecutive pairs of arcs: $L = \{(wu, uv) |\ w, u, v \in V,\ wu, uv \in A\}$. For the sake of shortness, members of L are called **pairs**. Throughout the paper by a **network code** we mean a pair $(\alpha, \mathbf{c})$ referring to local and global coefficients, respectively. The **local coefficients** of a network code is a function α on the pairs: $\alpha : L \to \mathbb{F}_q$. The **global coefficients** of a network code is a function $\mathbf{c} : A \to \mathbb{F}_q^k$ such that

$$\mathbf{c}(uv) = \sum_{wu \in A} \alpha(wu, uv) \mathbf{c}(wu)$$

for every arc $uv \in A, u \neq s$.

Assume that global coefficients $\mathbf{c}(sv) \in \mathbb{F}_q^k$ for every arc $sv \in A$ leaving s as well as a local coefficients α are given. From the acyclic property of the graph we get that these values uniquely determine global coefficients on every arc in the graph (see e.g. [4]).

Definition 2.2. A network code is **feasible** for receiver set T, if for every receiver node $t \in T$ the dimension of $\langle \mathbf{c}(vt) | vt \in A \rangle$ is k.

2.2 Failure protecting network codes

The notion of failure protection can be defined in several ways. We use the definition of [3] for failure protecting network codes, and our goal is to find a network code that remains feasible after a certain number of arc failures , that is, deleting any subset of the arcs the network code remains feasible on the remaining subgraph without altering local coefficients on failureless pairs.

Definition 2.3. Assume a network code $(\alpha, \mathbf{c})$ is given on a network. An **failure** is a subset of arcs $H \subseteq A$, and the network code $(\alpha_H, \mathbf{c}_H)$ resulting from $(\alpha, \mathbf{c})$ by H is setting all local coefficients to zero on pairs intersecting H (a pair (uv, vw) intersects H if $\{uv, vw\} \cap H \neq \emptyset$). If $(\alpha, \mathbf{c})$ is a feasible network code, we say that $(\alpha, \mathbf{c})$ **protects failure** H, if $(\alpha_H, \mathbf{c}_H)$ is also feasible. For a positive integer $d \in \mathbb{N}$ a network code is **d-failure-protecting**, if it protects any failure of size at most d. Similarly, given a set $\mathcal{H} \subseteq 2^A$ of possible failures, a network code is **$\mathcal{H}$-protecting** if it protects every failure $H \in \mathcal{H}$.

Note that such network codes enable a very fast, in fact instant recovery, because there is no need to inform the whole network about the failure, only a node with a failing entering arc needs to be able to recognize the failure.

3 Previous and new bounds

3.1 Related work

In [3], the existence of a protecting code over sufficiently large fields was characterized and a polynomial time algorithm was given by reducing the problem to simultaneous matrix completion. Given a failure H, it is easy to see that for the existence of an H-protecting network code it is necessary that there remain k arc-disjoint paths to every receiver node from s in $(V, A \setminus H)$. In [3], Harvey et al. showed that this is also sufficient, even for failure sets.

Theorem 3.1. *[Harvey, Karger, Murota [3]] There exists an $\mathcal{H}$-protecting network code (α, c) if and only if for every $H \in \mathcal{H}$ there are k arc-disjoint paths from s to every receiver in $(V, A \setminus H)$. Moreover, a protecting network code can be chosen over any field of size $q > |T||\mathcal{H}|$ in time $O(|T||\mathcal{H}|(m^3 \log m + |L|m^2))$.*

We can deduce the following theorem for the special case of d-protection.

Theorem 3.2. *There exists a d-failure protecting code if and only if $\lambda(s, t) \geq k + d$ for every receiver t. Such a code can be found over any finite field of size at least $|T|(\binom{m}{d} + \cdots + \binom{m}{0})$ in time $O(|T|(\binom{m}{d} + \cdots + \binom{m}{0})(m^3 \log m + |L|m^2))$.*

3.2 New bound

The main result of this paper is that the term m can be eliminated from sufficient field size bound for d-protection. Bahramgiri and Lahouti in [1] also gave similarly network size independent lower bounds for the required field size, but their algorithm used random network codes.

Theorem 3.3. *If a d-failure protecting network code exists, then such a network code can be found over any field of size $q > |T|(\binom{N}{d} + \cdots + \binom{N}{0})$, where $N = 3\binom{|T|}{2}(k + d)^3$. The running time of such an algorithm is $O(m^2|T|(k + d) + |T|(\binom{N}{d} + \cdots + \binom{N}{0})N^3 \log N)$.*

This lower bound on the field size may be much smaller for large graphs. The idea of the proof is based on a completely different topic called network encoding complexity. The proof will be described in 4.3, but first some results from the area of encoding complexity are presented, which we will use for the proof.

4 Network encoding complexity

4.1 Definitions and previous results

When constructing network codes in practice, one may notice that typically very few nodes perform actual coding, most nodes just forward one of the incoming messages on each outgoing arc.

Definition 4.1. Given a network code $(\alpha, \mathbf{c})$, a node v is called a **coding node** if there exists an outgoing arc vz such that there are at least two non-zero local coefficients $\alpha(uv, vz)$ and $\alpha(wv, vz)$ corresponding to vz.

The topic of network encoding complexity deals with the following fundamental question: Given a network coding problem, what is the minimum number of coding nodes for a feasible network code construction?

In [6], it was proved that the number of coding nodes can always be bounded by a value independent of the size of the network.

Theorem 4.2. *[Langberg, Sprintson, Bruck, [6]] Given a network such that $\lambda(s,t) \geq k$ for every $t \in T$, there exists a feasible network code over any finite field $\mathbb{F}_q$ with $q > |T|$ with at most $k^3\binom{|T|}{2}$ coding nodes.*

The proof of the theorem is based on a reduction of the original graph to an auxiliary graph with all internal nodes of degree two or three. Since we also use this auxiliary graph in our proof, we repeat its detailed construction in the following subsection.

4.2 Auxiliary graph construction

We describe a graph construction slightly different from that in [6]. We can assume that each receiver node is a sink. The construction has four steps. In the first step, each arc e is subdivided by two new nodes e^{in} and e^{out}. Then each internal node is substituted by a set of arcs as follows. Let v denote such a node with in-degree ρ and out-degree δ and with incoming and outgoing arcs $f_1, f_2, \dots, f_\rho$ and $e_1, \dots, e_\delta$, respectively. For each $1 \leq i \leq \rho$ and $1 \leq j \leq \delta$ we add node $w_{i,j}$ and arcs $(f_i^{out}, w_{i,j})$ $(w_{i,j}, e_j^{in})$ to the graph (see Fig. 1). Note that in this new graph nodes of the form e^{in} and e^{out} have out-degree and in-degree one, respectively. In the second step, if a node of the form e_j^{in} has in-degree more than two, the incoming arcs are substituted by a graph, where each node has in-degree at most two. Let $w_{1,j}, w_{2,j}, \dots, w_{\rho,j}$ denote the tails of arcs entering e_j^{in}. Then for each $2 \leq i \leq \rho - 1$, arc $w_{i,j}$ is subdivided by a new node z_i and the head of arc $(w_{i+1,j}, e_j^{in})$ is replaced by z_i, as shown in Figure 2. Similar procedure is made on nodes of the form e^{out} with out-degree more than two.

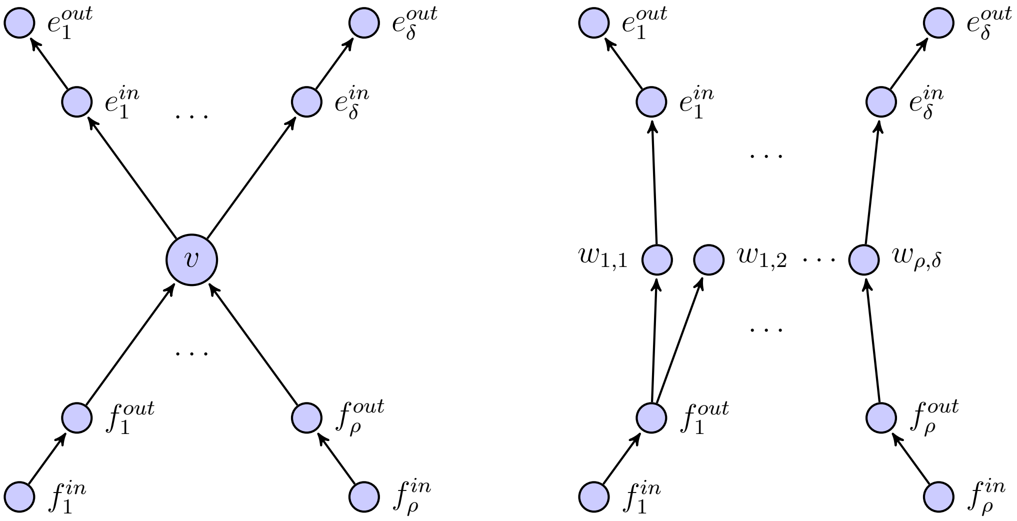

Figure 1: Substitution of node v in the first step.

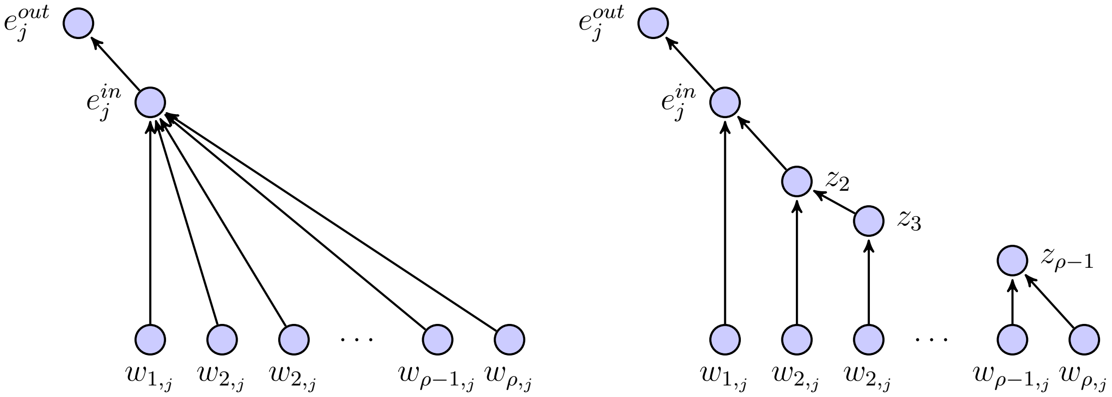

Figure 2: Substitution of arcs entering e_j^{in}.

The resulting graph after the two steps is denoted by $D'(V', A')$. Note that every internal node in D' has total degree at most three. We can observe the following relationship between paths in D and D'.

Proposition 4.3. *Let (f_i, e_j) be a pair in D. Then there is exactly one path P_i^j from f_i^{out} to e_j^{in} in D'. P_i^j is edge-disjoint from path P_k^l if and only if $i \neq k$ and $j \neq l$.*

In the third step, we omit arcs from D' as long as the connectivity requirements between s and the receivers are satisfied ($\lambda(s,t) \geq k$ for all $t \in T$). Let us denote the remaining graph $D''(V'', A'')$.

Finally, in the fourth step nodes with exactly one indegree and outdegree are replaced by a single arc as follows.

Definition 4.4. A **branch** of a graph D is a directed uv-path P in D such that $\rho(u) \neq 1$ and $\delta(v) \neq 1$ but all other in-degrees and out-degrees of nodes in P are one.

Note that every arc in a graph is covered by exactly one branch. Specifically, in D'' only nodes in $T + s$ or with total degree three can be endpoints of a branch. Let us substitute each branch of D'' by a single arc and let us denote the final graph by $D^*(V^*, A^*)$. Theorem 4.2 was proved by the following lemma.

Lemma 4.5. *[6] D^* has at most $\binom{|T|}{2}k^3$ internal nodes.*

For our proof we need to define a mapping ϕ from A'' to A^*: for an arc $e \in A''$, $\phi(e)$ is the arc corresponding to the branch containing arc e in D''. For a set of arcs $H \subseteq A''$, by abuse of notation, let $\phi(H) := \{\phi(e) | e \in H\}$.

4.3 Proof of Theorem 3.3

From Theorem 3.1 we get that for the existence of a d-protecting network code on D, the existence of $k + d$ arc-disjoint paths from s to each receiver is a necessary and sufficient condition. So we can construct the auxiliary graph $D^*(V^*, A^*)$ for $k + d$ paths. The idea is to find a d-protecting network code on D^* over a finite field, finally map it to an failure protecting code on D over the same field.

From Theorem 4.2 we get that $|A^*| \leq 3\binom{|T|}{2}(k + d)^3$. Since internal nodes have degree three, $|L| \leq 4|A^*|$, so setting $N = 3\binom{|T|}{2}(k + d)^3$ we get that $O(m^3 \log m + |L|m^2) \leq O(N^3 \log N + N^3) = O(N^3 \log N)$. Note that d-protection is equivalent to $\mathcal{H}$-protection, where $\mathcal{H}$ is the set containing all subsets of A^* of size at most d. Since $|\mathcal{H}| = \binom{|A^*|}{d} + \ldots + \binom{|A^*|}{0}$, by applying Theorem 3.1 to D^*, we get that there exists a d-protecting network code $(\alpha^*, \mathbf{c}^*)$ on D^* over any finite field of size at least $|T||\mathcal{H}|$, where $|\mathcal{H}|$ is a function of $|T|, k, d$.

Lemma 4.6. *A d-protecting network code $(\alpha^*, \mathbf{c}^*)$ on D^* can be transformed into a d-protecting network code $(\alpha, \mathbf{c})$ on D over the same finite field.*

For continuity we leave the proof of the lemma for the next subsection. Applying Lemma 4.6 for network code $(\alpha^*, \mathbf{c}^*)$ we get Theorem 3.3.

4.4 Proof of Lemma 4.6

We map $(\alpha^*, \mathbf{c}^*)$ one-by-one to D'' then D' and finally D, always maintaining d-protection and the field size. The existences of these mappings are proved by three claims. First we show that a network code on D' can be naturally transformed into one on D , and failure protection is preserved on arcs of type (e^{in}, e^{out}). For an arc set $F \subseteq A$, let us define $F' := \{(f^{in}, f^{out}) \in A' | f \in F\}$. Similarly, we define $F'' := F' \cap A''$.

Claim 4.7. *For a network code $(\alpha', \mathbf{c}')$ on D', there exists a network code $(\alpha, \mathbf{c})$ on D over the same field such that for every failure set F, if $(\alpha', \mathbf{c}')$ is F'-protecting then $(\alpha, \mathbf{c})$ is F-protecting.*

Proof. Let $\mathbf{c}(e) := \mathbf{c}'(e^{in}, e^{out})$. Observe from Figure 1 that arcs $(f_1^{in}, f_1^{out}), \ldots, (f_\rho^{in}, f_\rho^{out})$ form an s, e_j^{in}-cut in D'. Hence we have that $\mathbf{c}'(e_j^{in}, e_j^{out}) \in \langle\{\mathbf{c}'(f_i^{in}, f_i^{out}) | 1 \leq i \leq \rho\}\rangle$. If we set a local coefficient $\alpha(f, e)$ as the product of local coefficients of α' along path P_i^j, then $\mathbf{c}_F(e) = \mathbf{c}'_{F'}(e_j^{in}, e_j^{out})$ for every arc $e \in A \setminus F$ and the failure-protecting part of the claim follows. □

The following claim shows the relation between network codes on D'' and D'.

Claim 4.8. *A network code $(\alpha'', \mathbf{c}'')$ on D'' corresponds to a network code $(\alpha', \mathbf{c}')$ on D' such that for every failure $M' \subseteq A'$, if $(\alpha'', \mathbf{c}'')$ is M''-protecting then $(\alpha', \mathbf{c}')$ is M'-protecting.*

Proof. A network code on D'' can be regarded as a network code on D' such that $\mathbf{c}(e) = \mathbf{c}'(e)$ for every arc $e \in A''$ and $\alpha'(e, f) = \alpha(e, f)$ if $e, f \in A''$, whereas $\mathbf{c}'(e) = 0$ if $e \notin A''$ and α' set to zero on all pairs intersecting deleted arcs. □

Claim 4.9. *If there exists a d-failure protecting network code $(\alpha^*, \mathbf{c}^*)$ on D^*, then there exists a d-failure protecting network code $(\alpha'', \mathbf{c}'')$ on D'' over the same field.*

Proof. We set $\mathbf{c}''(e) := \mathbf{c}^*(\phi(e))$, and set all local coefficients on pairs of a branch of D'' to one. If two arcs e, f in A'' are on the same branch B, then $\mathbf{c}''_e = \mathbf{c}''_f$, and their failure corresponds to the same failure $\phi(e) = \phi(f)$ in A^*. So for every failure K in D'', $\mathbf{c}''_K = \mathbf{c}^*_{\phi(K)}$ hence the failure protection follows. □

Combining Claims 4.7, 4.8, 4.9 we get the proof of Lemma 4.6.

5 Conclusion

We have given a lower bound of $|T|(\binom{N}{d} + \ldots + \binom{N}{0}), N = 3\binom{T}{2}(k+d)^3$ on the required field size for a d-protecting network code, and an algorithm for such a code construction with running time of $O(m^2|T|(k+d) + |T|(\binom{N}{d} + \cdots + \binom{N}{0}))N^3 \log N)$.

Acknowledgment

This research was supported by grants (no. CNK 77780 and no. K 109240) from the National Development Agency of Hungary, based on a source from the Research and Technology Innovation Fund.

References

[1] H. Bahramgiri and F. Lahouti. Robust network coding against path failures. *Communications, IET*, 4(3):272–284, 2010.

[2] S. El Rouayheb, E. Soljanin, and A. Sprintson. Secure network coding for wiretap networks of type ii. *IEEE Transactions on Information Theory*, 58(3):1361–1371, 2012.

[3] N. J. A. Harvey, D. R. Karger, and K. Murota. Deterministic network coding by matrix completion. In *SODA*, pages 489–498, 2005.

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

[5] R. Koetter and M. Médard. An algebraic approach to network coding. In *IEEE/ACM Transactions on Networking*, volume 11, pages 782 – 795, Oct. 2003.

[6] M. Langberg, A. Sprintson, and J. Bruck. The encoding complexity of network coding. *IEEE/ACM Trans. Netw.*, 14(SI):2386–2397, June 2006.

[7] S.-Y. Li, R. Yeung, and N. Cai. Linear network coding. 49(2):371 –381, Feb. 2003.