# Robust Network Coding Using Information Flow Decomposition

Mohammad Rezagholipour
Department of Electrical Engineering,
K.N. Toosi university of Technology,
Tehran, Iran
Email: mrezagholipour@yahoo.com

Mahmoud Ahmadian
Department of Electrical Engineering,
K.N. Toosi university of Technology,
Tehran, Iran
Email: m_ahmadian@kntu.ac.ir

Mohammad R. Aref
Department of Electrical Engineering,
Sharif university of Technology,
Tehran, Iran
Email: aref@sharif.edu

*Abstract*-**In this paper, we propose a new approach to find a static solution for a multicast network problem. Our work relates the existence of a static network code to minimal subtree graphs of a given network. For a given multicast network there are different solutions. Sometimes some links of a network may be removed and then we will be encountered with link failure in the network. Therefore, to choose a specific satisfactory network code solution among different options, we must see how much the network code adds the network robustness against link failures. In this paper we extend the definition of the minimal subtree graph and replace it with a new definition of minimal subgraph and show that the static network code that is robust against all of solvable link failure patterns is a code that comes from solving all minimal subgraphs of a given network simultaneously.**

*Index Terms*-**Network coding, multicast, robustness, minimal subtree graph, link failure**

## I. INTRODUCTION

Network coding first proposed by Ahlswede, et. al. [1] has been improved to be an effective way to achieve the maximum flow in multicast networks, when one source node wants to transmit information to multiple sink nodes. An algebraic framework for network coding was developed by Koetter and Medard in [2] who translated the network code design to an algebraic problem.

In [3] Fragouli and Soljanin defined the minimal subtree graph on the basis of information flow decomposition which makes the analysis of network coding problem easier. Their approach is partitioning the network graph into subgraphs through which the same information flows. Processing, i.e., combining different flows, happens only at the border of these subgraphs, and the structure of the inside part of subgraphs does not play any role, and it is sufficient to know how subgraphs are connected and which receivers observe the information flow in each subgraph. Thus, it is possible to consider each subgraph as a new node and retain only the links connected to it. They showed that each subgraph is a tree with a source node or a coding point in its root. Therefore, they named the resulted graph as a subtree graph. Also they deleted some unnecessary parts of the subtree graph and retain the minimal subtree graph in the sense that the multicast property is hold. In this paper we first extend their work and then use it for finding a static network code which is robust [4] against all solvable link failure patterns.

For a given multicast network [5], [6] there are different coding solutions. Sometimes some links of a network may be removed by failure and then we will be encountered with link failure in the network. Therefore, one of the objectives of selecting a specific network code solution among different options is to maximize the robustness provided by this solution. We are interested in knowing the situation of every links of the network graph, however in the definition proposed for minimal subtree graph, lots of links and nodes of network may become invisible because we put them in a new node and there will not be any access to them. So we define *minimal subgraph*, which is an extension of minimal subtree graph [3]. This enables us to reduce the dimensionality of a network inspite of having enough knowledge about each network link situation from the failure point of view. Thus, we can see all of the links of the network graph associated to the desired multicast process, even the links in one subtree with the same information flow. This allows us to know which of the network links may fail without loosing the multicast property [3] and the network coding problem is still solvable; by comparing the network which has encountered link failure with minimal subgraphs of the network.

In this work we propose a new minimal subgraph, and show that the static network code which is robust against all of solvable link failure patterns is a code obtained through solving all minimal subgraphs of a given network, simultaneously.

This work was partially supported by Iranian NSF under Grant No.84.5193 and partially supported by Iran Telecommunication Research Center (ITRC) under Grant No. 500/9120.

## II. PROBLEM FORMULATION

WIOPT 2008, 1st – 3rd Apr 2008, Berlin, Germany.

DOI 10.4108/ICST.WIOPT2008.3236

A network may be represented mathematically by a directed graph $G = (V, E)$. Our work is restricted to the multicast network problem in which a source node $s$ is connected to $h$ nodes $s_1, s_2, \ldots, s_h$ with $h$ links with unit capacity so that it seems that the network has $h$ unit rate information sources $s_1, s_2, \ldots, s_h$. In the rest of this paper instead of the source node $s$, we denote these $h$ secondary sources as the $h$ unit rate information sources of the network unless it is specified. We assume that all of the network links have unit capacity. Also the network has $N$ receivers $R_1, R_2, \ldots, R_N$; and the number of links of the min-cut between the sources and each receiver node is $h$ [7]. The $h$ sources multicast information simultaneously to all $N$ receivers at rate $h$. Our work is restricted to acyclic delay-free networks. As we are concerned with the multicast problem there are lots of options for choosing a set of $h$-link-disjoint paths between the $h$ source nodes and each receiver $R_j$. This variety of options affects the complexity of the network code. Our interest is arbitrary choosing subgraph $G'$ of $G$ consisting $hN$ paths $(S_i, R_j)$, $1 \le i \le h$, $1 \le j \le N$; so that the other parts of the network graph $G$ which don't involve in the subgraph $G'$ don't play any role in solving the problem.

This approach to network coding has advantages such as reduction of dimensionality of the network code design problem, bounding the network code alphabet size [2], [8]; and finding the coding nodes in which the network coding operation is necessary [3], [9], [10]; that makes the analysis of dense and complicated networks easier.

Our work represents a new application of minimal subgraph and also opens a new window for obtaining robust network code. A practical challenge which sometimes happens is that links in a network may fail. Then we encounter to the following questions:

*1)* Under which failure patterns a network coding problem is still solvable?

*2)* Is there any network coding solution which resists against all of the solvable failure patterns?

*3)* How can we find that solution?

Question 1 will be answered by a definition and the second is answered by Koetter and Medard [2]. For the 3rd question, we propose a new way for finding the desired solution, which is more rapid and less complicated than that proposed in [2].

## III. ROBUST NETWORK CODE VIA SIMULTANEOUS SOLVING OF MINIMAL SUBGRAPHS

We extend the definition of minimal subtree graph [3] to *minimal subgraph* in order to study the network robustness. Here we represent the algorithm IV.1 proposed in [3] but with extending the result to minimal subgraph.

*Algorithm 1*

Minimal subgraph ( $(S_i, R_j)$, $1 \le i \le h$, $1 \le j \le N$ )

$$\gamma = (V_\gamma, E_\gamma) \leftarrow \bigcup_{\substack{1\le i\le h\\ 1\le j\le N}} L(S_i, R_j)$$

$C \leftarrow$ coding points of $\gamma$

$\varepsilon \leftarrow$ links of $\gamma$ terminating in C

$$\text{for each } e \in \varepsilon \begin{cases} \text{if } (V_\gamma, E_\gamma \setminus \{e\}) \text{ satisfies the multicast property} \\ \text{then } \begin{cases} E_\gamma \leftarrow E_\gamma \setminus \{e\} \\ \gamma \leftarrow (V_\gamma, E_\gamma) \\ C \leftarrow \text{ coding points of } \gamma \\ \varepsilon \leftarrow \text{edges of } \gamma \text{ terminating in C} \end{cases} \end{cases}$$

minimal subgraph= $\gamma = (V_\gamma, E_\gamma)$

return $\gamma$.

Denoted by $\{(S_i, R_j), 1 \le i \le h\}$ a set of $h$-link-disjoint paths from the sources to the receiver $R_j$. Since, for a network code, we have eventually to describe which operations each node in $G'$ has to perform on its inputs, it is more clear to work with the graph $\gamma = \bigcup_{\substack{1\le i\le h\\ 1\le j\le N}} L(S_i, R_j)$, where, $L(S_i, R_j)$ denotes the line graph of the path $(S_i, R_j)$. That is, $L(S_i, R_j)$ is the graph with vertex set $E(S_i, R_j)$ in which two vertices are joined if and only if they are adjacent as links in $(S_i, R_j)$. Also we assume that the line graph contains a node corresponding to each of the $h$ sources. We refer to these nodes as *source nodes*. Each node with a single input link merely forwards its input symbol to its output links. Each node with two or more input links performs a coding operation i.e. linear combining of its input symbols, and forwards the result to its output links. Such nodes are named as *coding points*.

A link failure pattern can be identified with a binary vector $f$ of length $|E|$ such that each position in $f$ is associated with one link in $G$. If a link fails we assume that the corresponding value in $f$ equals one, otherwise the entry in $f$ corresponding to the link equals zero. Given a network $G$ and a link failure pattern $f$ it is straightforward to consider the network $G_f$ obtained by deleting the failed links. We call the link failure pattern $f$ solvable if the network $G_f$ is still solvable. We are interested in static solutions, where the network is oblivious to the particular failure pattern. As described in [2] we can formulate our linear network model by a transfer matrix $M = [\lambda_{i,j}]$ describing the relationship between the input and the output vectors of the network whose coefficients $\lambda_{i,j}$ are elements of a finite field; i.e. $M$ is the system transfer matrix of the linear network coding problem.

We say that a network solution is static under a set $F$ of link failure patterns if there exist solutions for the network under any link failure pattern $f \in F$, with the same elements $\lambda_{i,j}$.

*Lemma 1*

*1)* For a minimal subgraph $G_{minimal\text{-}i}$ of the graph $G$, there exists a set $F_i$ of failure patterns, so that the obtained network code by solving $G_{minimal\text{-}i}$, is a static code for all graphs $G_f$ ( $f \in F_i$ ).

*2)* Suppose that there exist $L$ minimal subgraphs for a given network graph $G$. Corresponding to each minimal subgraph $G_{minimal\text{-}i}$, there exists a set $F_i$ of failure patterns so that the obtained network code through solving $G_{minimal\text{-}i}$ is a static code for all graphs $G_f$ ($f \in F_i$ ). Thus we have sets $F_1, F_2, \ldots, F_L$ .

*Proof:*

*1)* Let $M_f$ and $M_{minimal\text{-}i}$ be the system transfer matrix of the network graphs $G_f$ and $G_{minimal\text{-}i}$, respectively, and the set $F_i$ consists of all solvable link failure patterns, such that $G_{minimal\text{-}i}$ is a subgraph of every $G_f$. Therefore owing to the existence of a solution for $G_{minimal\text{-}i}$, there will exist a solution for $G_f$ which involves $G_{minimal\text{-}i}$ plus some other links of the network graph $G$. That is why we can choose the entries of $M_f$ belonging to those links which are common in both $G_f$ and $G_{minimal\text{-}i}$ equal to the entries of $M_{minimal\text{-}i}$ and the entries belonging to those other links in matrix $M_f$, equal to zero (This is the same as choosing local encoding kernels of those other links, equal to zero [5] ). Therefore the network code solution for $G_{minimal\text{-}i}$ would be a network code solution for $G_f$.

*2)* Follows from 1.

*Theorem 1*

If $F$ is a set consisting of all solvable link failure patterns $f$; for every solvable network $G_f$, where $f \in F$, there will exist at least a minimal subgraph such that the solution for that minimal subgraph is also a solution for $G_f$.

*Proof:*

$G_f$ is a graph which is obtained by deleting the failing links correspond to the failure pattern $f$, and is still solvable. Therefore, it has the multicast property. Thus, we encounter one of the two situations:

*1)* $G_f$ is exactly one of those $L$ minimal subgraphs and then it would be solvable and the theorem is proved.

*2)* At least one of those $L$ minimal subgraphs, is a subgraph of $G_f$. Since the graph $G_f$ is obtained by deleting some links from graph $G$ and has still the multicast property; therefore by deleting some other links of it (until removing any more link would violate the multicast property), we obtain a minimal subgraph. Therefore, if at least one of those $L$ minimal subgraph say $G_{minimal\text{-}i}$ is a subgraph of $G_f$, owing to the existence of a solution for $G_{minimal\text{-}i}$, there will exist a solution for $G_f$ which involves $G_{minimal\text{-}i}$ plus some other links of the network graph $G$. That is why we can choose the entries of $M_f$ belonging to those links which are common in both $G_f$ and $G_{minimal\text{-}i}$ equal to the entries of $M_{minimal\text{-}i}$ and the entries belonging to those other links in matrix $M_f$, equal to zero (This is the same as choosing local encoding kernels of those other links, equal to zero [5] ).

A direct consequence of the above theorem is given in the following corollary.

*Corollary*

Let network $G$ be given and have $L$ minimal subgraphs. For every minimal subgraph $G_{minimal\text{-}i}$, there exists a set $F_i$ consisting some link failure patterns $f$, such that $\bigcup_{i=1}^{L} F_i = F$. In other words for every link failure pattern $f$ ($f \in F$) there exists a set $F_i$ corresponds to $G_{minimal\text{-}i}$, where $f \in F_i$.

*Proof:*

We have shown in theorem 1 that, for every link failure pattern $f$ ($f \in F$), there exists a minimal subgraph $G_{minimal\text{-}i}$ (where the solution of $G_{minimal\text{-}i}$ is also a solution for $G_f$ ); and have shown in lemma 1 that, corresponding to every minimal subgraph, there exists a set $F_i$ of some failure patterns. Therefore for every solvable link failure pattern $f$, there exists a set $F_i$, such that $f \in F_i$. For this reason it is obvious that $\bigcup_{i=1}^{L} F_i = F$ .

*Lemma 2 ( [2] )*

Let $\mathbb{F}[X_1, X_2, \ldots, X_n]$ be the ring of polynomials over an infinite field $\mathbb{F}$ in variables $X_1, X_2, \ldots, X_n$. For any nonzero element $f \in \mathbb{F}[X_1, X_2, \ldots, X_n]$ there exists an infinite set of $n$-tuples $(x_1, x_2, \ldots, x_n) \in \mathbb{F}^n$ such that $f(x_1, x_2, \ldots, x_n) \neq 0$ .

*Theorem 2*

Let a linear network code and a set $F$ of all solvable link failure patterns be given. There exists a common, static solution to the network problem $G_f$ for all $f \in F$ obtained by solving equation $g(\underline{\lambda}) \neq 0$, where $g(\underline{\lambda}) = \prod_{j=1}^{N} \prod_{i=1}^{L} g_{f_i,j}(\underline{\lambda})$; and $f_i$ is a solvable link failure pattern such that $G_{f_i} = G_{minimal\text{-}i}$ .

*Proof:*

Let $G$ has $L$ minimal subgraph $G_{minimal\text{-}i}$ and i=1, 2,…, $L$ . For every $G_{minimal\text{-}i}$ there exist a set $F_i$ of some link failure patterns, such that the solution of $G_{minimal\text{-}i}$ is so accepted for $G_f$ ( $f \in F_i$ ). Let $g_{f_i,j}(\underline{\lambda})$ be determinant of transfer matrix representing the connection between source node $s$ and receiver node $R_j$, after that the failure pattern $f_i$ is employed to network graph $G$. we consider the product $g(\underline{\lambda}) = \prod_{j=1}^{N}\prod_{i=1}^{L} g_{f_i,j}(\underline{\lambda})$. By lemma 2, we can find an assignment of numbers to $\underline{\lambda}$ such that $g(\underline{\lambda})$ and hence every single determinant $g_{f_i,j}(\underline{\lambda})$ evaluate to a nonzero value simultaneously. It follows that regardless of error pattern in $F$ the basic multicast requirements are satisfied. Therefore this solution is a static solution for all solvable link failure patterns.

Inspiring the above theorem, here we bring an algorithm for finding a static network code. Let a network $G$, with source node $s$ and receiver nodes $R_1$, $R_2$,…, $R_N$ ; and a set $F$ of all solvable link failure patterns be given.

*Algorithm 2: A static network code*

*Step 1)* Find all $L$ minimal subgraphs corresponding to the given network $G$.

*Step 2)* each minimal subgraph $G_{minimal\text{-}i}$ corresponds to a link failure pattern $f_i$ such that $G_{minimal\text{-}i} = G_{f_i}$. find $f_i$ where i=1, 2, …, $L$ .

*Step 3)* Perform the product $g(\underline{\lambda}) = \prod_{j=1}^{N}\prod_{i=1}^{L} g_{f_i,j}(\underline{\lambda})$, where $g_{f_i,j}(\underline{\lambda})$ is the determinant of transfer matrix representing the connection between source node $s$ and receiver node $R_j$ , after the failure pattern $f_i$ is employed to $G$.

*Step 4)* Solve the equation $g(\underline{\lambda}) \neq 0$. The obtained coefficients $\lambda_{i,j}$ perform the desired static network code solution for the network graph $G$, which is robust against all solvable link failure patterns.

Using algorithm 2 for finding a static network code solution is faster and less complicated than the way proposed by Koetter and Medard [2]. They used the product $g(\underline{\lambda}) = \prod_{j=1}^{N}\prod_{i=1}^{|F|} g_{f_i,j}(\underline{\lambda})$ at equation $g(\underline{\lambda}) \neq 0$, to obtain the static network code solution. But as we mentioned before, we use the product $g(\underline{\lambda}) = \prod_{j=1}^{N}\prod_{i=1}^{L} g_{f_i,j}(\underline{\lambda})$ at equation $g(\underline{\lambda}) \neq 0$.

By comparing these two different $g(\underline{\lambda})$ it is easy to see that we reduced the number of multiplication operations from $|F|$ to $L$ which is less than $|F|$. Hence we reduced the number of multiplication operations by the ratio $L / |F|$ .

Based on this reason, our algorithm needs finite fields with smaller size than Koetter and Medard's. Because the coefficients vector $\underline{\lambda}$ which comes from solving $g(\underline{\lambda}) \neq 0$, must satisfy every equation $g_{f_i,j}(\underline{\lambda}) \neq 0$ for different $i$'s and $j$'s simultaneously, where, the number of equations in our algorithm is less than Koetter and Medard's. Hence for finding convenient coefficients $\lambda_{i,j}$, we must search a finite field with smaller size than the Koetter and Medard's finite fields.

In addition, due to the reduction of the multiplications number, the coefficients $\lambda_{i,j}$ which are chosen from such a field must satisfy less equations $g_{f_i,j}(\underline{\lambda})$ for all $i$'s and $j$'s simultaneously, which leads to a faster method than the way proposed in [2].

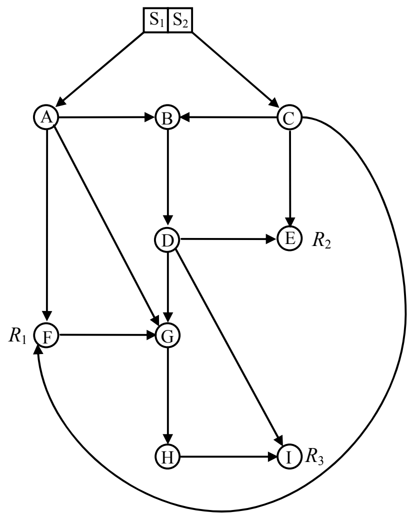


Fig. 1. Network with two unit rate sources and three receivers.

To illustrate the above issue, we consider the example of a network with two sources and three receivers in Fig.1. For this network graph there is nine solvable link failure patterns, i.e. $|F|$=9. But the number of minimal subgraphs is three.

## IV. CONCLUSION

In this paper, we introduced a faster method with lower complexity for finding a static network code resistant against all solvable link failure patterns of a network. We have shown that the static network code solution can be obtained by solving all minimal subgraphs of a given network, simultaneously.

## REFERENCES


[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. "Network Information Flow", *IEEE Transactions on Information Theory*, Vol. 46, pp. 1016-1024, 2000

[2] R. Koetter and M. Medard. "An Algebraic Approach to Network Coding", *IEEE/ACM Transactions on Network Coding*, Vol. 11,pp. 782-795, 2003.

[3] C. Fragouli and E. Soljanin, "Information Flow Decomposition for Network Coding," *IEEE Transactions on Information Theory*, Vol. 52, No3, March 2006.

[4] S. Jaggi, P. Sanders, P. A. Chou, M. E_ros, S. Egner, K. Jain, and L. Tolhuizen., "Polynomial Time Algorithms for Network Code

Construction”. *IEEE Trans. Information Theory*, 51:1973-1982, June 2005.
[5] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, “Network Coding Theory,” *Foundation and Trends in Communications and Information Theory*, vol. 2, no. 4 and 5, pp. 241–381, 2005.
[6] A. Rasala-Lehman and E. Lehman, "Complexity Classification of Network Information Flow Problems", *41st Annual Allerton Conference on Communication Control and Computing*, Oct. 2003.
[7] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, 1962.
[8] S.-Y. R. Li, Raymond W. Yeung, and Ning Cai. "Linear Network Coding". *IEEE Transactions on Information Theory*, 49(2):371–381, 2003.
[9] M. Langberg, A. Sprintson and J. Bruck, ”The Encoding Complexity of Network Coding,” ETR063, California Institute of Technology. http://www.paradise.caltech.edu/papers/etr063.pdf
[10] M. Kim, c. w. Ahn, M. Medard, M. Effros, “On Minimizing Network Coding Resources: An Evolutionary Approach,” *Second Workshop on Network Coding, Theory, and Applications(NetCod)*, April 2006. http://www.mit.edu/~minkyu