International Journal of Computing and Digital Systems

A New Methodology for Network Coding

Amaal S. Abd Elhameed, Yasmeen. A. Fahmy and Magdy S. El-Soudani

Dept. of Electronics and Communications Engineering, Cairo University, Egypt

e-mail: amalsamir@eece.cu.edu.eg , yasfahmy@eece.cu.edu.eg, melsoudani@ieee.org

Received 27 Oct 2013, Revised 8 Nov 2013, Accepted 13 Nov 2013, Published 1 Jan 2014

Abstract: To maintain network reliability and performance, it must be protected against two common problems; link and node failures. Network protection codes (NPC) were proposed to protect operational networks against these failures, where encoding and decoding operations of such codes were developed over binary and finite fields. Finding network topologies, practical scenarios, and limits on graphs applicable for NPC are of interest. In this paper, we investigate a new method to represent the network coding topology. This method is equivalent to the conventional representation used, but it is more easier. This method on the contrary of the conventional representation can be extended for larger networks and can be used efficiently to prevent data losses. Several applications such as security and multicast can also make use of this method.

Keywords: Network coding; The G-method.

I. Introduction

As networks have become the backbone of life, the failure of single link or node can cause a loss of huge amount of information, which may lead to a disaster.
Therefore network connections are designed to face such failures by using several techniques such like adding external network resources, or serving network resources to be as a backup circuits for the recovering process. The recovery process should also be as quick as possible to reduce network delay [1, 2].
In network coding techniques, instead of a simple forward operation, we allow the relay nodes to encode the incoming packets from all sources into one output packet. The output packet is then forwarded to all destinations. On the receiver side, the packets are decoded by linearly solving them together [3-5].
This approach provides some benefits like minimizing network delay, maximizing network throughput, and allowing the nodes to achieve the optimal performance in [6,7].

This paper is organized as follows. In Section II, the algebraic method to represent the network coding is shortly explained for sake of completeness. For more details, the reader is referred to [8]. In Section III, we illustrate the new method and give clear relations between both methods and clearly state the main advantages of this proposed representation. Section IV presents the use of the new representation method in achieving the network coding applications; *multicasting*, *network resilience*, and *data security*. Finally, section V concludes the paper.

II. ALGEBRAIC REPRESENTATION METHOD

In this section, the conventional method used to represent the network topology is presented and then the main limitations are stated.

Starting by defining the network model as G, where G = (V,E) is a graph (network) with the set of vertices' V and the set of edges $E \epsilon V \times V$. We assume that each edge has unit capacity, and allow parallel edges. Consider a node $S \epsilon V$ that wants to transmit information to a node $R \epsilon V$. From the MCMR (min-cut max-rate) theorem, if the min-cut between S and R equals h, then the information can be send from S to R at a maximum rate of h.

$$\text{Max rate} \leq h \tag{1}$$

Equivalently, there exist exactly h edge disjoint paths between S and R [9]. The theorem says that intermediate nodes can perform linear operations, namely, additions and multiplications over a finite field F_q, which we can refer to as linear network coding.

Koetter and Medard [6] presented an algebraic framework for network coding. They put an algebraic form for the network model.

They defined $\chi(\upsilon)$ to be the input random processes to the source S and $\rho_{total}(\upsilon)$ denotes the output at υ. The random process transmitted through link e by Y(e). A node υ can observe random processes Y (e') for all e', where e' is the edge at the end of node υ and Y (e') is the random process transmitted through e'.

They defined the network in layers such that each layer is represented by number of equations, so the network is represented by several equations as

$$Y(e)=\sum_{l=1}^{\mu(\upsilon)} \alpha_{l,e}\, \chi(\upsilon) + \sum_{e':head(e')=tail(e)} \beta_{\{e',e\}}\, Y(e') \tag{2}$$

where the coefficients $\alpha_{l,e}$ and $\beta_{e',e}$ represent the network topology and they are elements of F_2^m. Where F_2^m is the finite field with 2^m elements.

The output $\rho_{total}(\upsilon)$ at any node υ is

$$\rho_{total}(\upsilon,j) = \sum_{e':head(e')=\upsilon} \varepsilon_{e',j}\; Y(e') \tag{3}$$

where the coefficients $\varepsilon_{e',j}$ are elements of F_2^m.

Considering F_2^m a *linear network* then we can give a *transfer matrix* that describes the relation between the input vector χ and the output vector ρ_{total}, which is G such that

$$\rho_{total} = \chi\, G \tag{4}$$

G is a transfer matrix whose coefficients $\alpha_{l,e}$ and $\beta_{e',e}$ and $\varepsilon_{e',j}$ are elements of F_2^m . G can be represented as

$$G = B\,(I-F)^{-1} A^T \tag{5}$$

Where F is the adjacency matrix of the graph G with elements $F_{i,j}$ given as

$$F_{i,j} = \begin{cases} \beta_{e_i,e_j} & head(e_i) = tail(e_j) \\ 0 & \text{otherwise} \end{cases}$$

B is defined as

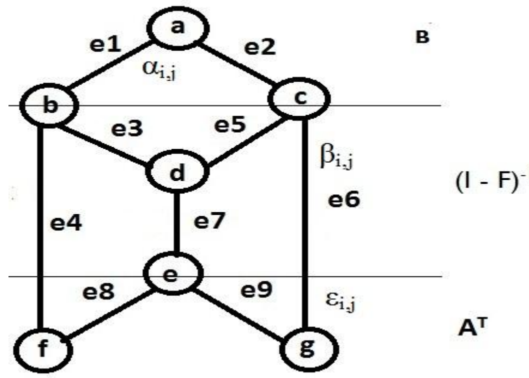
$$B_{i,j} = \begin{cases} \alpha_{l,e_j} & x_i = X(tail(e_j),l) \\ 0 & \text{otherwise} \end{cases}$$

A is defined as

$$A_{i,j} = \begin{cases} \varepsilon_{e_j,l} & z_i = Z(head(e_j),l) \\ 0 & \text{otherwise} \end{cases}$$

Consider the network given in Fig.1, we will make use of the above mentioned equations to describe the network.

Figure1. The butterfly network defining each edge on it



$$Y(e_1)= \alpha_{1,e1} X_1 + \alpha_{2,e1} X_2$$
$$Y(e_2)= \alpha_{1,e2} X_1 + \alpha_{2,e2} X_2$$
$$\rho_{total}(f,1)= \rho_1= \varepsilon_{e4,1}\, y_{e4}+ \varepsilon_{e8,1}\, y_{e8}$$
$$\rho_{total}(g,2)= \rho_2 = \varepsilon_{e6,2}\, y_{e6} + \varepsilon_{e9,2}\, y_{e9}$$

Then matrix A and B can be defined as

$$B = \begin{bmatrix} \alpha_{1,e_1} & \alpha_{2,e_1} \\ \alpha_{1,e_2} & \alpha_{2,e_2} \end{bmatrix}$$

$$A = \begin{bmatrix} \varepsilon_{e_4,1} & 0 & 0 & 0 \\ 0 & \varepsilon_{e_6,2} & 0 & 0 \\ 0 & 0 & \varepsilon_{e_{84},1} & 0 \\ 0 & 0 & 0 & \varepsilon_{e_9,2} \end{bmatrix}$$

From the above equations, the transfer matrix G will be

$$G = B \begin{bmatrix} \beta_{e_1,e_4} & \beta_{e_1,e_3}\beta_{e_3,e_7} & \beta_{e_1,e_3}\beta_{e_3,e_7} & 0 \\ 0 & \beta_{e_2,e_5}\beta_{e_5,e_7} & \beta_{e_2,e_5}\beta_{e_5,e_7} & \beta_{e_2,e_6} \end{bmatrix} A^T \tag{6}$$

Later C.Fragouli et al. unified all the network coefficients α_i, β_i and ε_i into one coefficient α_i [8]. Thus, the network topology is represented by α_i as in Fig.2 .

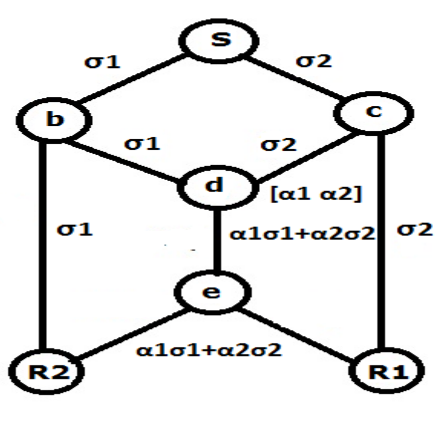


Figure 2. Network with conventional representation

As shown in Fig.2, In normal operation, we assume that different information data flow from one source S that is connected to all receivers R_j through the network connections.
S performs a linear combination on the input information data $\tau = [\tau_1 \tau_2 \ldots \tau_m \ldots \tau_M]^T$ and transfers them into $\sigma = [\sigma_1 \sigma_2 \ldots \sigma_n \ldots \sigma_N]^T$. This combination is performed by the input matrix B of size N ×M where,

$$\sigma = B\,\tau \qquad (7)$$

The different symbols (σ_n) start to flow through the network until they reach the destinations R_j . Then R_j solve the system of linear equations to get the required τ_m. Let ρ_j be the vector that represent the symbols on the last edge on the path (S, R_j), and A_j be the matrix whose rows are the coding vectors of the last edge on the path (S, R_j). Then the linear equations are presented as:

$$\rho_j = A_j B\,\tau \qquad (8)$$

Each receiver R_j has a corresponding mapping matrix A_j . In case of link failure: the link is mapped into the different A_j matrices by 0's in the corresponding positions. With the help of the direct link, each R_j solves the linear combination of the transmitted information τ with the available links to get the missing τ_m [10,11]. For our example in Fig.2, the two receivers observe the linear combinations of source symbols defined by the matrices

$$A_1 = \begin{bmatrix} 1 & 0 \\ \alpha_1 & \alpha_2 \end{bmatrix} \quad , \quad A_2 = \begin{bmatrix} 0 & 1 \\ \alpha_1 & \alpha_2 \end{bmatrix}$$

So at receiver R_1 and R_2, we get

$$\rho_1 = \begin{bmatrix} 1 & 0 \\ \alpha_1 & \alpha_2 \end{bmatrix} \quad ,$$

$$\rho_2 = \begin{bmatrix} 0 & 1 \\ \alpha_1 & \alpha_2 \end{bmatrix} \quad .$$

III. THE NEW REPRESENTATION METHOD

In our new method to represent the network topology, we do not consider a one matrix transformation as in the conventional representation. Each network layer is presented by a transformation matrix.

For each layer l with input links K_l and output links N_l, we choose a different sub G_l matrix of size $K_l \times N_l$ to define the relation between the K_l links and the N_l links. It worth noting that the outputs of layer l -1 are the inputs to layer l, then we have $K_l = N_{l-1}$ for all l = 0, 1, ...L-1. The matrix G0 define the relation between the input information data τ and the output σ from the source S as $\sigma^T = \tau^T$ G0, it can be easily related to the input matrix defined above as $B = G_0{}^T$.

We define G_{NC} the matrix of dimensions $M \times N_{L-1}$ as the Network Coding matrix, it is given by:

$$G_{NC} = G_0\ G_1 \ldots G_l \ldots\ldots G_{L-1} \qquad (9)$$

We can then deduce $\rho_{total} = [\rho_0\ \rho_1\ \rho_2 \ldots\ \rho_{NL-1}]$ the received vector of all receivers as:

$$\rho_{total} = \tau^T\ G_{NC} \qquad (10)$$

A. The relation between different models

In this subsection, the relation between the G matrices of the proposed representation model and the other models is clarified.

The G-matrix model is a simple method to model larger networks. Where we do not have to draw the whole network to trace all links to get the data from the last edge on the path (S, R_j) for all receivers as in the conventional method. Instead, we divide the network into layers and represent the layers with number of sub G matrices, which are easier to deal with to get the data for all receivers.

There is a difference between our model and Koetter model in [6] such that he represents the network's layers by equations to get its transfer matrix; however, in the proposed layered method, we represent each layer with a separate transfer matrix sub-G which gives us the ability to change the shape of the internal network by changing the elements of the sub- G matrices. In the Koetter method, we have to calculate the inverse of the intermediate matrix $(I\text{-}F)^{-1}$ to get the overall G matrix, while in the layered method, it only depends on the matrices multiplication operations.

The relation between the transfer matrix of both methods is:

G in Koetter's model is

$$G = B(I - F)^{-1}A^T$$

while G in our model is

$$G = G_s\ G_{int}\ G_{con};$$

such that B is like G_s as both represent the encoding matrix that encodes the information into the network, and the same for both A and G_{con} as both connect the transmitted information through the network to the receivers. As for $(I - F)^{-1}$ and G_{int}, they are equivalent to each other.

B. Examples

We illustrate how the proposed G-method is used for network representation by the next two examples.

1) First Network Example: For the network shown in Fig.3, we have M = 2 information data to transmit, the source has N = 8 outputs

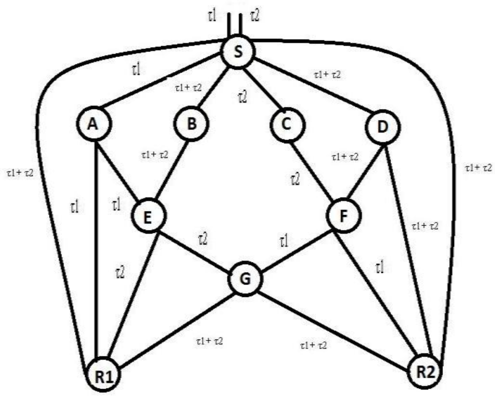


Figure 3. The butterfly network model

When we apply the proposed G-method, We model the network as layers and represent each layer with its equivalent sub G matrix given as:

$$G_0 = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix},$$

$$G_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$G_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

Finally, $G_{NC} = G_0\ G_1\ G_2\ G_3$

$$= \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

The final outputs of layer L-1 (the inputs to the receivers) can then be given by (10) as:

$$\rho_{total} = [\,\tau_1+\tau_2\ \ \tau_1\ \ \tau_2\ \ \tau_1+\tau_2\ \ \tau_1+\tau_2\ \ \tau_1\ \ \tau_1+\tau_2\ \ \tau_1+\tau_2\,] \quad (11)$$

Multiplying element by element by Gcon given for this example as

$$G_{Con} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (12)$$

we get the received data:

$$R_1 = [\,\tau_1+\tau_2\ \ \tau_1\ \ \tau_2\ \ \tau_1+\tau_2\ \ 0\ 0\ 0\ 0\,],$$
$$R_2 = [\,0\ 0\ 0\ 0\ \ \tau_1+\tau_2\ \ \tau_1\ \ \tau_1+\tau_2\ \ \tau_1+\tau_2\,] \quad (13)$$

2) Second Network Example: In this second example, we use a much larger network, Fig.4 and apply both methods on it. Applying the G-method's model, we use the following matrices to define each layer. We find that the G-method is more suitable to use in larger networks than the A-method.

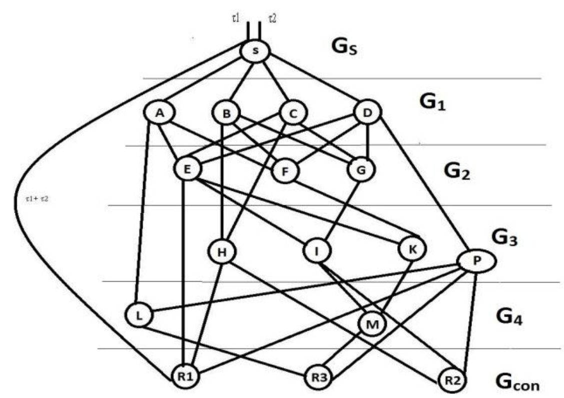


Figure 4. A general network model

We use the following matrices to define each layer:

$$G_s = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix},$$

$$G_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \alpha_4 & \alpha_7 \\ 0 & 0 & 1 & 0 & 0 & \alpha_1 & 0 & \alpha_8 \\ 0 & 0 & 0 & 1 & 0 & \alpha_2 & \alpha_5 & 0 \\ 0 & 0 & 0 & 0 & 1 & \alpha_3 & \alpha_6 & \alpha_9 \end{bmatrix},$$

$$G_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_{10} & \alpha_{10} & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_{11} & \alpha_{11} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_{13} & 0 & 1 & \alpha_{14} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_{15} \end{bmatrix},$$

$$G_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \alpha_{18} & 0 \\ 0 & \alpha_{17} & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_{19} & 0 \end{bmatrix},$$

Finally, $G_{NC} = G_0 \; G_1 \; G_2 \; G_3 \; G_4$

$$= \begin{bmatrix} 1 & \varphi_1 & \varphi_2 & 1 & \alpha_{11} & \pi_1 & 1 & \alpha_{11} & \pi_2 & 1 \\ 1 & \alpha_{17} & \varphi_3 & 1 & \varphi_4 & \pi_3 & 1 & \varphi_4 & \pi_4 & 1 \end{bmatrix}$$

Where

$\psi_1 = (\alpha_{16} + \alpha_{17})$

$\psi_2 = (\alpha_4 + \alpha_5 + \alpha_6)$

$\psi_3 = (\alpha_5 + \alpha_6)$

$\psi_4 = (\alpha_{10} + \alpha_{11})$

$\pi_1 = (\alpha_{12} \, (\alpha_2 + \alpha_3) + \alpha_{13} \, (\alpha_4 + \alpha_5 + \alpha_6))$

$\pi_2 = (\alpha_{18} \, (\alpha_{12} \, (\alpha_2 + \alpha_3) + \alpha_{13} \, (\alpha_4 + \alpha_5 + \alpha_6) \,) +$
$\quad \alpha_{19} \, (\alpha_{14} \, (\alpha_7 + \alpha_9) + \alpha_{14} \, (\alpha_4 + \alpha_5 + \alpha_6)) \,)$

$\pi_3 = (\alpha_{12} \, (\alpha_1 + \alpha_2 + \alpha_3) + \alpha_{13} \, (\alpha_5 + \alpha_6))$

$\pi_4 = (\alpha_{18} \, (\alpha_{12} \, (\alpha_1 + \alpha_2 + \alpha_3) + \alpha_{13} \, (\alpha_5 + \alpha_6) \,) +$
$\quad \alpha_{19} \, (\alpha_{14} \, (\alpha_5 + \alpha_6) + \alpha_{14} \, (\alpha_8 + \alpha_9)) \,)$

The connection matrix is given by

$$G_{con} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

The R_j values (neglecting the zeros) are:

$R_1 = [(\tau_1 + \tau_2), (\tau_1), (\tau_1 + \tau_2), (\tau_1)]$,
$R_2 = [(\tau_1), (\tau_1 + \tau_2), (\tau_1 + \tau_2)]$,
$R_3 = [(\tau_2), (\tau_2), (\tau_1 + \tau_2)]$. (14)

IV. MULTICAST

A. Multicast using The G-method

The ability to change the intermediate **G** matrices can also be applied to achieve the **Multicast** case [10].

For example, in Fig.4 if the requirements are given as:

_ R_1 receives τ_2.
_ R_2 receives τ_1.
_ R_3 receives τ_1 and τ_2.

This can be achieved by changing the functions of nodes **E**, **G** and **F** as shown by the following sub-G matrices:

$$G_s = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix},$$

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$G_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$G_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

The network code matrix, $G_{NC} = G_0 \, G_1 \, G_2 \, G_3 \, G_4$, is then given by

$$G_{NC} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

and with the use of the connection matrix

$$G_{con} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

we get the R_j values

$R_1 = [(\tau_2), (\tau_2), (\tau_2), (\tau_2)]$,
$R_2 = [(\tau_1), (\tau_1), (\tau_1)]$,
$R_3 = [(\tau_2), (\tau_2), (\tau_1 + \tau_2)]$.

Which are the main requirements stated earlier.

B. Network Resilience in Multicasting using The G-method

Network resilience is the ability to provide and maintain an acceptable level of normal operation in the face of failures.

The failure of single link or node can cause a loss of huge amount of information and throughput to decrease especially in multicasting, where the number of resources are limited, therefore the throughput decrement will be large. Network coding helps in lost data recovery due to performing linear operation at the nodes.

The **G-method** enables the data recovery using two ways:

1) The redundancy links at each receiver, which enables the receiver to recover any link failure[11-13].

2) The ability to change the sub-G matrices, which enables us to avoid the failed link, or the failed node.

In the following sub-sections, we show how we can use the G-method in network resilience.

1) Error Representation: There is an easy way to model ***Link*** and ***Node*** failures in the G-method, instead of tracing back the network to get the exact values of the A matrices. A link failure in the link (n, k) between layer l□1 and layer l is modeled by inserting an identity matrix E of size $N_l \times N_l$, between the G_{l-1} and G_l matrices, with a zero entry in the failure position (n, n).

Node failure is considered multiple link failures, The E matrix has zero entries in all the positions of the output links of the failed node.

Back to our examples, by using any of the two representations, we can get the same results for Link and Node failures. In the first network, Fig.3, for any single link failure of the 18 link, the first receiver will be able to overcome the failure, while the second receiver fail to receive the data in only one case. In the second example Fig.4, the first and second receivers are able to overcome 24 single failures out of 27, while the third receiver is able to overcome 25 single failures.

As for Node failure, we always get the data at the first receiver, while the failure affect the second and third receivers in only one node. These results are achievable with both techniques, but easier to analyze using the proposed representation.

2) Link Failure Recovery: The G-method gives us the ability to control the function of each node, therefore nodes may forward the summation of the incoming data (xor different inputs) or they can forward a specific input data to achieve a pre-requisite or to overcome a failure. For example in Fig.4, R_3 cannot overcome two failures, the failure of link *D-R*$_3$, in this case it will only receive τ_1. And the failure of link *S-A*, it will then receive $\tau_1 + \tau_2$ and will not be able to solve any message. We can solve this by changing the G_2 matrix such as to change the function of node *F*. The new matrix is then:

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix},$$

and this results in

$$G_{NC} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

After this modification, R_3 will have both $\tau 1$ and $\tau 2$.

To overcome the failure of link *E-K*, we may change the function of node *F* and keep the network normally operating.

In this case G_2 change to be:

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix},$$

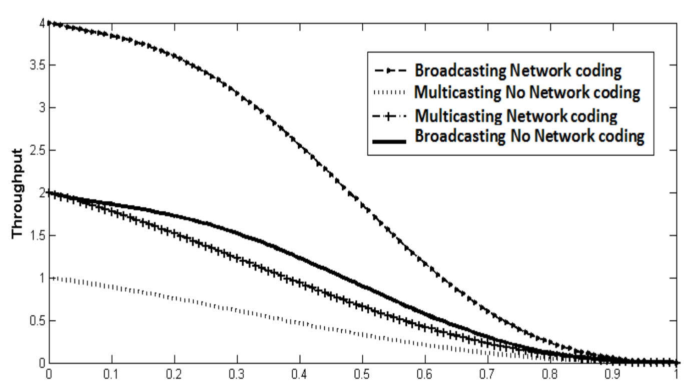
and this results in the same original G_{Nc} matrix.

3) Simulation Results: We simulate a multicast network, where there are two input information τ_1 and τ_2 transferred through the network and two receivers R_1 and R_2. The information are transferred such that receiver R_1 gets only τ_1 and receiver R_2 gets only τ_2.

Matlab is used to model the network using the two different methods; the Forwarding method and the Network Coding method. We assume constant number of time slots and the number of packets increases for each time slot, where two new packets are added in each time slot. We also assume that link failure means that the link has zero data. The probability of failure changes from ’0’ to ’1’ and the link failure probability depends on the probability of failure and takes its value randomly either ’0’ or ’1’, where ’0’ represents the link failure and ’1’ represents a good link. We use **Matlab** to simulate the network overall throughput at different probabilities of link failure.

Fig.5 shows the overall throughput of both receivers using the Forwarding method and the Network Coding method. It compares the network throughput of the broadcasting and multicasting networks, we find that the multicasting throughput with network coding is nearly equal to the broadcasting throughput with the forwarding method. This advantage of network coding is important especially for the multicasting network, where the network has lower number of resources, as a result of the division between different receivers to transfer different data, these lower resources are affected by any failure causing the data loss at any of the receivers. Therefore, we can say that network coding doubles the network throughput over the forward methods. It also maintains the network throughput, however the number of failures occurred through the data transmission.

Figure 5. Comparing The Broadcasting and Multicasting Throughput



With and Without Network Coding

C. Data Security in Multicasting using The G-method

Security is specially needed when multicasting different sessions to different receivers. where each receiver has different information than the other receivers and each one must not know the data of the others. Unsecured network coding may results in the ability of some receivers to know the data of other receivers, which is not acceptable in multicasting networks [14].

We use the G-method to represent the network by dividing the network into layers and represent each layer with a G_i, $1 \leq i \leq L$ matrix and use the network coding to transmit the information through the network nodes and links.

We can solve the problem of low security of the network coding, by using different G matrices such that we use G matrix for each information, so if we have M information symbols, we will have M G matrices such that each G_i; $1 \leq i \leq M$ matrix for transmitting each symbol, where we transmit the symbols in different paths. By this way, we ensured that each of the intermediate nodes won't be able to receive all symbols of information and know the data transmitted through the network, and we ensured that each receiver will see only its data and won't see the other receivers' data.

For example in Fig.3, using the above G_{NC}, both receivers will have both τ_1 and τ_2.

If we want the receive R_1 to have only τ_1 and the receive R_2 to have only τ_2, then we will use two different G_{NC} one for transmitting τ_1 for R_1 and one for transmitting τ_2 for R_2 without letting any intermediate node to have both information symbols as following:

$$G_{s1} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$G_{11} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$G_{21} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_3 & \alpha_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_4 & \alpha_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$G_{31} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_5 & \alpha_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_6 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

Finally, $G_{NC1} = G_{s1}\ G_{11}\ G_{21}\ G_{31}$

$$= \begin{bmatrix} 1 & 1 & \alpha_1 & \alpha_5\alpha_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$G_{s2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix},$$

$$G_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$G_{22} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_1 & \alpha_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_2 & \alpha_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$G_{32} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_5 & \alpha_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

Finally, $G_{NC2} = G_{s2}\ G_{12}\ G_{22}\ G_{32}$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_6\alpha_3 & \alpha_4 & 1 & 1 \end{bmatrix}$$

The final outputs of layer L-1 (the inputs to the receivers) can then be given as:

$$\rho_{total} = [(\tau_1)\ (\tau_1)\ (\tau_1)\ (\tau_1)\ (\tau_2)\ (\tau_2)\ (\tau_2)\ (\tau_2)] \qquad (15)$$

The <u>main advantage</u> of this method is that it secures the data by sending them in different paths. Also by using the redundancy links input to each receiver, it ensured that the data won't be lost due to any failure in the network.

However, this method is vulnerable to eavesdropping attack. For a small network with low number of nodes per layer, some of the intermediate nodes of the layers will be able to receive all the information symbols transmitted over the network. Therefore the attacker will have the ability to know the whole information by putting different malicious nodes among the intermediate nodes with probability that one of these nodes is one of the nodes which will receive all the transmitted information symbols.

V. CONCLUSION

In this paper, we proposed a new method to model larger network for the purpose of network coding. We call this the *G-method* to distinguish it from the algebraic network coding method. This new representation can be easily used to reconfigure the network without knowing the whole network connections, only with the knowledge of the total number of nodes N_t and the number of transmitted messages M and can also be used to overcome some link or node failures.

We used the *G-method* in achieving three different applications of network coding; multicasting, network resilience, and security.

In multicasting, we used the proposed a model to achieve the network multicasting , where we set the different sub G matrices to allow the multicasting between the different receivers.

One of the main benefits of network coding is increasing the throughput, which was proved in the simulation results using the *G-method* to model the network. We proved that the throughput using network coding is double the throughput using forwarding methods for both broadcasting and multicasting.

Also, we used the *G-method* to secure the transmitted data over the multicasting networks, where we used different transfer matrix G_{NC} to transmit the information through different paths.

REFERENCES

[1]. S. Li, R. Yeung, and N. Cai, "Linear Network Coding," IEEE Trans. Inform. Theory, vol. 49, no. 2, pp. 371–381, February, 2003.

[2]. N. Cai and R. Yeung, "Secure Network Coding," IEEE Trans. Inform. Theory, pp. 323–325, June, 2002.

[3]. T. chan and N. Cai, "Theory of Secure Network Coding," IEEE Trans. Inform. Theory, vol. 99, no. 3, pp. 421–437, March, 2011.

[4]. R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network Information Flow," IEEE Trans. Inform. Theory, vol. 46, pp. 1204–1216, August, 2002.

[5]. T. Ho, D. R. Karger, M. Medard, and R. Koetter, Network Coding from a Network Flow Perspective," IEEE Trans. Inform. Theory, vol. 441, July, 2003.

[6]. R. Koetter and M. Medard, "An Algebraic Approach to Network Coding," IEEE/ACM Trans. Networking, vol. 11, pp. 782–795, October, 2005.

[7]. N. Cai and R. W. Yeung, "Network Coding and Error Correction," in Proc. the 2002 IEEE. Information Theory Workshop, Bangalore, India, October, 2002, pp. 119–122.

[8]. C. Fragouli and E. Soljanin, "Network Coding Applications, Foundations and Trends in Networking," vol. 2, no. 2, pp. 135–269, 2007.

[9]. A. E. Kamal and S. A. Aly, "Network Protection Codes: Providing Self healing in Autonomic Networks Using Network Coding," in Proc. IEEE Global Telecommunications Conference, Honolulu, Hawaii, USA, November, 2009.

[10]. E. Soljanin, P. Gupta, and G. Kramer, "Network Coding for Efficient Network Multicast," in Bell Labs Technical Journal - Enabling Science and Technology, 2009, vol. 14, no. 3.

[11]. S. A. Aly, A. E. Kamal, and A. I. Walid, "Network Protection Design Using Network Coding," in Information Theory Workshop (ITW), Cairo, Egypt, January, 2010, pp. 1–5.

[12]. S. A. Aly and A. E. Kamal, "Network Protection Codes Against Link Failures Using Network Coding," in Proc. IEEE Global Telecommunications Conference, New Orleans, USA, November, 2008, pp. 1–6.

[13]. H. Yao, S. Jaggi, and M. Chen, "Network coding tomography for network failures," *IEEE Transactions on Information Theory*, 58(9), pp. 5922-5940, Sept 2012.

[14]. Peng Zhang, Yixin Jiang, Chuang Lin, Patrick P. C. Lee, and John C. S. Lui,"ANOC: Anonymous Network-Coding-Based Communication with Efficient Cooperation," *IEEE Journal on Selected Areas in Communications (JSAC)*, 30(9), pp. 1738-1745, Oct 2012.

Amaal Samir received the B.Sc with Honors from Cairo University in 2008, the MSc. in 2013 in computer networks, and all from the faculty of Engineering at Cairo University. She was appointed as a Research assistant, at Cairo University, from September 2008 to September 2013.

Dr. Yasmine Fahmy received the B.Sc with Honors from Cairo University in 1999, where she graduated as top of her class, the MSc. in 2001, and the PhD. in 2005, all in telecommunications, and all from the faculty of Engineering at Cairo University. She was appointed as a Research assistant, at Cairo University, from September 1999 to September 2005, then she was appointed as an Assistant Professor.

Dr. Fahmy held the position of an Associate Professor at Cairo University since March 2013. Her research areas include modern coding techniques for wireless communications systems and MIMO systems. Since 2009, she has been with the Center of Wireless Studies (CWS), Cairo University.

Magdy El-Soudani (SM 82, M83, 92) Professor of communications at the Electronics and Electrical Communications Engineering Department, Faculty of Engineering, Cairo University since 1995. His main fields of research include channel coding (Error correcting Techniques) , Network Coding and Security systems. More than fifty papers have been published in specialized technical journals or presented at international conferences.