Packet Output and Input Configuration in a Multicasting Session Using Network Coding

José Márquez[1*], Ismael Gutiérrez[2], Sebastián Valle[1] and Melanis Falco[1]
[1] System and Computer Department, Universidad del Norte
Km. 5 Puerto Colombia Highway, Barranquilla - Colombia
[e-mail: jmarquez@uninorte.edu.co]
[2] Math Department, Universidad del Norte
Km. 5 Puerto Colombia Highway, Barranquilla - Colombia
[e-mail: isgutier@uninorte.edu.co
*Corresponding author: José Márquez

Received February 9, 2018; revised June 21, 2018; accepted August 19, 2018;
published February 28, 2019

Abstract

This work proposes a model to solve the problem of Network Coding over a one-session multicast network. The model is based on a system of restrictions that defines the packet flows received in the sink nodes as functions of the outgoing flows from the source node. A multicast network graph is used to derive a directed labeled line graph (DLLG). The successive powers of the DLLG adjacency matrix to the convergence in the null matrix permits the construction of the jump matrix Source-Sinks. In its reduced form, this shows the dependency of the incoming flows in the sink nodes as a function of the outgoing flows in the source node. The emerging packets for each outgoing link from the source node are marked with a tag that is a linear combination of variables that corresponds to powers of two. Restrictions are built based on the dependence of the outgoing and incoming flows and the packet tags as variables. The linear independence of the incoming flows to the sink nodes is mandatory. The method is novel because the solution is independent of the Galois field size where the packet contents are defined.

Keywords: Multicast network, Linear Network Coding, Linear Constraint System, Network Coding Solution, Directed Labeled Line Graph

The authors would like to thank Research, Development and Innovation Management of the Universidad del Norte (DIDI) for their support in presenting this work.

http://doi.org/10.3837/tiis.2019.02.012 **ISSN : 1976-7277**

1. Introduction

Traditional routing mechanisms allow intermediate nodes to execute storage-and-forward actions; therefore, each packet (the element of the vector space $\mathbb{F}_q^n$, where $\mathbb{F}_q$ is the finite field with q elements and $q = 2$) is a copy of a previously received packet in an input link to the node. In multicast systems, intermediate nodes can receive packets over input links, besides they can be duplicated and forwarded to multiple output links. However, this mechanism is prone to traffic congestion.

However, Network Coding (NC) [1–3] allows each router, which acts as a node in the network, to carry out the mixing of its incoming packets by computing a linear combination, not only in the source and sinks nodes, but also in the intermediate nodes. **Fig. 1** shows a node (router) with three incoming packet flows and two outgoing flows, which are a linear combination of the incomers. When NC is used in transmissions, especially multicast transmissions, packets can be forwarded, mixed (coded) at intermediate nodes, and decoded to obtain the original packets at the sink nodes [4]. Therefore, NC theory shows that the transmission of information in a multicast system can be improved and surpass the paradigm of routing and data replication.

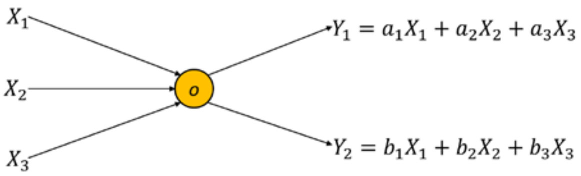

X_1
X_2
X_3
o
$Y_1 = a_1X_1 + a_2X_2 + a_3X_3$
$Y_2 = b_1X_1 + b_2X_2 + b_3X_3$

Fig. 1. Computing outgoing packets from incoming packets

According to [5], "*Network Coding means that bits in information flows do not have to be delivered as a commodity; these can be mixed as desired, as long as the receiving computers have received enough evidence or clues to reconstruct the original packets of the sending computer*."

Fig. 2 (in [4] and [6]) shows a network consisting of directed links with the same capacity of B bps. Without intermediate routing nodes, the source computers s_1 and s_2 send a packet stream to the receiving computers t_2 and t_1, respectively. In turn, the source computer s_1 attempts to send packets to computer t_1, and computer s_2 attempts to send packets to computer t_2. Observation shows that the outgoing packets of s_1 reach t_1 only through the path $s_1 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow t_1$, and the outgoing packets of s_2 reach t_2 only through the path $s_2 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow t_2$. It is clear that the link (3,4) is shared and acts as a bottleneck for simultaneous transmission.

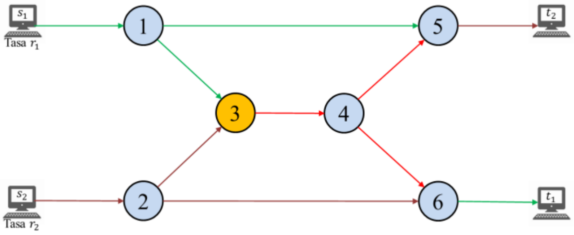


Fig. 2. Two unicast sessions in contention for the link (3,4) [4]

If the nodes that make up the **Fig. 2** network only route the information they receive, then link $(3,4)$ must be shared between the two unicast sessions, which only allows the following set of transmission rates $\{(r_1, r_2)|r_1, r_2 \geq 0, r_1 + r_2 \leq B\}$ and these are interpreted graphically in **Fig. 3(a)** [7]. It is not possible for any of the unicast sessions to establish a communication stream with a rate greater than B. However, if the network nodes can implement NC, specifically on node 3 of the shared link $(3,4)$, then both sessions can reliably communicate at rate B, which permits the set of transmission rates $\{(r_1, r_2)|0 \leq r_1 \leq B, 0 \leq r_2 \leq B\}$ as shown in **Fig. 3(b)** [7]. Therefore, the region of **Fig. 3(b)** comprises all of the possible flow rates, and this is called the capacity region for these sessions [4] [7]. For both unicast sessions, rate B is achieved by mixing the information flows in node 3 using the XOR operation. By applying the XOR operation again, the original flows are obtained in nodes 5 and 6. In **Fig. 2**, the blue and purple line links carry the original information flows, whereas the red links carry the mixed flow.

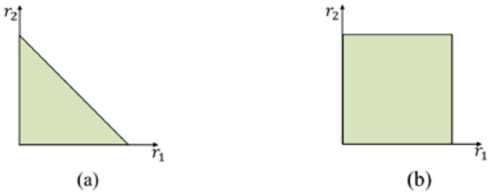


Fig. 3. The ranges of reachable transfer rates for **(a)** traditional routing and **(b)** routing with NC [4]

According to [8], the current communications networks share the same fundamental operating principle. Regardless of whether packets are sent over the Internet or signals on a telephone network, the information is transported like vehicles traveling along a highway. Just as cars share the road, data flows share the resources of the network. However, the information packets themselves, like individual cars, are separate.

Currently, communication networks treat data packets as unmodifiable units; that is, data packets produced at the source are routed through the network until they reach a sink. In this process, each packet remains intact. The NC paradigm, which was initially proposed in [1], breaks with this previous assumption. It suggests that instead of merely forwarding data, nodes can combine several incoming packets into one or more outgoing packets, which makes it unnecessary to deliver the actual packets produced by the source node to the sink nodes. The sink nodes can obtain the original packets through a process of decoding the received packets.

This work proposes a system of linear equations (constraints) whose solution determines the ordering of the r-packets over the vector space $\mathbb{F}_q^n$ in the outgoing links from the source node, where r corresponds to the maximum flow of a multicast session. This ordering ensures that the packets are correctly delivered to the sink nodes. Packets can be sent from the source node in the original format (single packets) or in linear combinations of single packets (combined packets). Also, combined packets can be created through inner nodes of the network. These packets, which are defined over $\mathbb{F}_q^n$, arrive at the sink nodes where they must be linearly independent so that the system has a solution.

The importance of this work is that the most relevant constraints derived from the solution define each incoming flow in each sink as a linear combination of some of the outgoing flows in the source node. The resulting linear combinations are expressed in terms of the codes representing the packets. The packets are tagged as variables based on a code system over the vector space $\mathbb{F}_2^r$, where the constraints are set with coefficients over $\mathbb{F}_2$.

This work can neither be extended directly to a network with losses or errors in the packets, nor when there are broken links. However, for future studies with transport protocols and error control, along with routing protocols for updating the routes, they can use the results achieved from this study for the most efficient delivery of the packets. Currently, large Internet providers include redundancy at three levels to mitigate these problems. The first is at the router level, which can have more than one high capacity processor installed. The second level is the connection of more than one link between a pair of routers. Finally, there is the redundancy of destination routers from an origin with a high traffic load. An example of a redundancy application is the Virtual Router Redundancy Protocol (VRRP) [29].

Knowledge of the topology is a prerequisite for the objective proposed in this work. An algorithm must run to determine the paths that lead from the source node to the set of sink nodes before the application of the proposed model. The algorithm would demand a high computational cost when determining the topology that favors the use of Network Coding due to the unknown and complex nature of the global network through which the multicast routing would be conducted. The determination of the multicast network has become one of the most complex tasks for the construction of a solution with Network Coding [19]. However, after the algorithm defines the topology, the overhead would be subject to the convergence times involved in the entry and exit of the nodes, and the breakdown and recovery of the links. These times are within a few seconds with current processor technologies.

Two topologies with coding nodes were utilized to show the application of the explained model. The topologies do not have a specific name. The main advantage of these topologies is that they show the simultaneous sending of packets from the source node, the construction of linear combinations in the coding nodes, and the decoding of packets in the sinks. The main disadvantage is that in the case of implementation, there should be a synchronization of the packets sent at the exit and arrival to the routers to achieve reception and decoding in the sinks.

This paper is organized as follows. Section 2 is a review of the studies related to the proposal. Section 3 explains the principles of construction of the network model. A detailed explanation of the proposed problem is included in Section 4. Section 5 describes the representation of the network model in matrices and the classification of the links. Section 6 discusses the construction of the solution bases and the algorithms that support it. Section 7 shows how to obtain the constraints of the model from the exposed bases. Section 8 provides examples of the implementation of this proposal. Finally, the conclusion and recommendation of future work are presented in Section 9.

2. Related Work

The initial theoretical work in NC was developed by Ahlswede *et al*. [1] and was modeled with a directed graph $G = (V, E)$ with links without noise and with capacities of a symbol of a field $\mathbb{F}_q$ per unit of time. They showed that a source node, $s \in V$, can send the same information to a set of sink nodes, $T = \{t_1, t_2, \dots, t_d\} \subset V$, where $s \notin T$, at a maximum data rate r.

Ahlswede *et al*. demonstrated that the problem with a multicast session is an underutilization of resources by restricting the use of the network nodes to only the execution of the routing function. Their work also demonstrated that the multicast capability, $r(s, T)$, or the maximum rate at which a source node can transmit common information to a set of sink nodes, is determined by the lower value of the maximum flows from the source node s to each of the sink nodes $t \in T$. That is:

$$r(s, T) = min_{t \in T} MaxFlow(s, t). \tag{1}$$

The above result is known as information flow theorem, which is considered a generalization of the classic max-flow-min-cut theorem for commodity flow [9] and was tested independently by Elías *et al.* [10] and Ford and Fullkerson [11] [12] in 1956.

NC demonstrates that it is possible to increase performance and gain an advantage over traditional routing mechanisms. Also, whereas routing mechanisms cannot achieve multicast capacity, it can be achieved through NC.

Li *et al.* [3] demonstrated that linear coding functions in the inner nodes are enough to achieve the maximum multicast capacity and thus define Linear Network Coding (LNC). LNC considers messages as packets consisting of elements over a finite field $\mathbb{F}_q$ [13], which can be represented by a binary vector of length $log_2(q)$ bits [6]. The coding function that generates a message emerging from a node is a simple linear combination of incoming messages to the node over the finite field $\mathbb{F}_q$. Likewise, linear operations on incoming information permit decoding in the sink nodes.

Koetter and Médard [2] [14] showed that linear solutions exist for solvable multicast networks with an alphabet of some finite field whose size is a power of two and is as large as $r|T|$. Also, they demonstrated how to find the coefficients of linear coding and decoding functions by searching for the values of the variables of a non-zero polynomial. However, for the solution of the multicast problem, the structure they proposed produces a polynomial time algorithm. Authors in [15–17] showed how to find the linear coding and decoding coefficients for a multicast session through an algorithm that runs in polynomial time over a finite field of maximum size $|T|$.

The authors of [18] proposed a significant improvement to the performance of end-to-end multicast transmission through the replication and forwarding of data by overlay nodes instead of sending them through end systems. In their work, they determined that overlay nodes have the total capacity to encode and decode data at the message level using LNC. They also proposed a distributed algorithm to construct a directed acyclic graph (DAG) corresponding to the multicast network in which NC is applied. This algorithm duplicates end-to-end performance in several cases.

The NC-based multicast routing algorithm proposed in [19] aims to find the path-clusters between the source node and each sink. The path-cluster corresponds to the r paths of disjoint links between the source node and each sink in a multicast system. Through the labeling algorithm, it is possible to find the maximum flow r between any two nodes in a network [20] [21] and also obtain the r disjoint paths for each sink node in a multicast session.

In [22], a routing and coding scheme is presented to achieve maximum packet transport in a multicast session by determining and combining the paths of the multiple maximum flows through overlapping links. The authors propose that only the links and nodes in the combined maximum flows will be used to construct the coding and routing scheme.

Based on the advantages of the shortest path Dijkstra algorithm and NC, the proposed routing algorithm [23] has a higher performance and load balancing capacity than the shortest path distribution tree algorithm.

3. Principles of Model-Building

3.1 Communications Network Model

This work studies multicast transmission (wired or wireless) with the purpose of sending the same information from a source node to all sink nodes at the same data rate (bandwidth). This communications network is modeled with a directed, acyclic and lossless (entirely reliable) link graph $G = (V, E)$, where V represents the set of nodes (routing only or routing/coding), and E is the set of directed links that join the nodes. Let $i, j \in \mathbb{N}$, then it is established that

$\forall e \in E$, it is represented as the ordered pair $e = (i, j)$, where $i, j \in V$ are ordered topologically, and this indicates that $i < j$. The flow of information goes from node i to node j, and delays in transmission are not considered. The origin and the destination node of a link $e = (i, j)$ is denoted by $o(e) = i$ and $d(e) = j$, respectively. In simple multicast communication, the sender or source node $s \in V$ can transmit the same information to a set of sink nodes $T \subseteq V$, where $s \notin T$.

$\Gamma_I(i)$ is defined in [2] as the set of links that enter node $i \in V$, and $\Gamma_O(i)$ as the set of links that originate in i. Formally, this is written as:

$$\Gamma_I(i) = \{e \in E \mid d(e) = i\}. \tag{2}$$

$$\Gamma_O(i) = \{e \in E \mid o(e) = i\}. \tag{3}$$

$\delta_I(i)$ is also defined as the degree or number of links entering i, whereas $\delta_O(i)$ is the degree or number of links that emerge from i: formally, $\delta_I(i) = |\Gamma_I(i)|$ and $\delta_O(i) = |\Gamma_O(i)|$.

Each link $e \in E$ is associated with a non-negative number $C(e)$, which is called the capacity of e. A node i can send information through a link $e = (i, j)$, at a maximum rate of $C(e)$ bits per unit of time. The bit stream (called a random process in [2] [24]) is transmitted through link e and is denoted by f_e o f_{ij}. Also, at node i, the flows $f_{e'}$ enter through each $e' \in \Gamma_I(i)$. In general, the stream of the bits f_e, transmitted through link $e = (i, j) \in \Gamma_O(i)$ will be a function of all of the flows $f_{e'}$, assuming that $e' \in \Gamma_I(i)$.

3.2 Maximum Transmission Rate

The model aims to send the same information from the source node to each sink node at the maximum possible data rate. We let $r(s, T)$ (or simply r) denote the rate at which the same information can reliably be transmitted to each sink node $t \in T$. It is observed that the maximum transmission rate or bandwidth, $r(s, T)$, has as an upper bound related to the minimum of the maximum flows between s and each $t \in T$, which indicates that $r(s, T) \leq min_{t \in T} MaxFlow(s, t)$. Likewise, in [25], the MaxFlow-MinCut theorem proved that the minimum cut, s-t, is equal to the maximum flow for the unicast case, i.e., $r(s, T) \leq min_{t \in T} MinCut(s, t)$. This maximum transmission rate is only achievable in a network where multicast communication is performed using NC, and this is demonstrated by [1].

In the proposed model, the maximum transmission rate, $r = min_{t \in T} MaxFlow(s, t)$, between source node s and set T of the sink nodes in a multicast session corresponds to the maximum number of packets generated in the source node s, which is the same number of packets delivered in each of the nodes $t \in T$. The generated and delivered r packets are l symbols length (bits that are over the field $\mathbb{F}_2$). Therefore, the unit of measure of r in bits (symbols) per unit of time (as mentioned in earlier sections) is no longer considered.

3.3 Symbols, Codes and Fields

The directed links of G can reliably carry packet p, where p is a vector of length l bits set over the finite field $\mathbb{F}_q, q = 2^m$, and m is the number of bits per symbol over the finite field. The number L of symbols of length m bits constituting the packet is determined by:

$$L = \left\lceil \frac{l}{log_2|\mathbb{F}_q|} \right\rceil = \left\lceil \frac{l}{log_2(2^m)} \right\rceil = \left\lceil \frac{l}{m} \right\rceil. \tag{4}$$

For example, given a network with a packet p of 12 bits long and 3 bits per symbol (they are over the field $\mathbb{F}_{2^3}$), the following values are deduced:

The length of packet l is 12 bits. The number of m bits per symbol over the finite field $\mathbb{F}_q$ is 3 bits per symbol. The number of 3-bit symbols over the finite field $\mathbb{F}_q$ is $|\mathbb{F}_q| = 2^3 = 8$ 3-bit symbols. The number of symbols in the packet is $L = \left\lceil \frac{12\ bits}{3\ bits/symbol} \right\rceil = 4$ symbols over the field $\mathbb{F}_q$. It is observed that the packet has a length of 12 bits, $p \in \mathbb{F}_q^4$, with $q = 2^3$. All of the packets generated and transmitted through the network links will be over the vector space $\mathbb{F}_{2^3}^4$, and a total of $\left|\mathbb{F}_{2^3}^4\right|$ different packets can be generated.

A sample of these packets is shown below with the 3-bit symbols grouped in parentheses:

$$p_1 = [(001)(010)(100)(110)]$$
$$p_2 = [(010)(000)(110)(110)]$$
$$p_3 = [(100)(110)(001)(101)]$$

3.4 Vector Spaces and Packets

Packets are classified as simple or combined. Simple packets are the original packets generated by the client's computer placed before the source node and obtained (through linear decoding) in the $|T|$ sink nodes, whereas combined packets are the result of linear combinations that occur along the multicast transmission from the source node s to any of the sink nodes in T.

The source node s receives the r simple packets that determine the maximum flow of the multicast session from the sending computer, and each content of a simple packet p is defined over $\mathbb{F}_2^l$. For example, if $l = 7$, the packets generated and received by s belong to the vector space $\mathbb{F}_2^7$ and are 7 bits long.

In the model, each packet corresponds to an information unit constituted by l bits and each network link is defined by a bandwidth equal to the information unit. All of the links in the model have the same capacity.

Let $c(p)$ be the code of packet p; the simple and combined r packets built along the transmission to the T-nodes are represented (or tagged) by codes belonging to the set:

$$\wp = \{c(p) | c(p) \in [1 : 2^r - 1]\}. \tag{5}$$

The codes of simple packets shape the $\wp_u$ set, such that $\wp_u \subset \wp$, where:

$$\wp_u = \{c(p) | c(p) = u_i = [0^{i-1}, 1, 0^{r-i}], i = 1, \dots, r\}. \tag{6}$$

The vector u_i is the i^{th} vector of the canonical basis of $\mathbb{F}_2^r$. The number of codes in $\wp$ that can be handled in a multicast session of maximum flow r is $|\wp| = 2^r - 1$, and the number of codes corresponding to the simple packets is $|\wp_u| = r$.

In addition, the simple packet codes (or tags) are also represented by mnemonics corresponding to an alphabet $\Sigma = \{A_1, A_2, \dots, A_r\}$ and are related following the topological ordering of the alphabet symbols as specified in the following bijective function:

$$\begin{gathered} g : \wp_u \longmapsto \Sigma \\ g(u_i) = A_i \end{gathered} \tag{7}$$

From function g, it follows that $\Sigma = \{A_i | A_i = g(u_i),\ u_i \in \wp_u\}$, and $|\Sigma| = r$.

For example, if $r = 3$, the sets $\wp, \wp_u$ and Σ are:

$\wp = \{001, 010, 011, 100, 101, 110, 111\}$, $\wp_u = \{001, 010, 100\}$ and $\Sigma = \{A, B, C\}$, where $g(001) = A$, $g(010) = B$, and $g(100) = C$.

From the above, the codes representing the r packets that can be transmitted from the source node s, in a multicast session are nonzero elements of the vector space $\mathbb{F}_2^r$. That is, packets are distinguished by their representative codes and not by their contents. Likewise, if the mnemonics or codes that identify two or more packets are different, then so are their contents.

In each coding/routing node, the outgoing packet corresponds to a linear combination of the incoming packets and uses as global coding coefficients [24], the symbols (bits) with a value of 1 belong to the field $\mathbb{F}_2$. This linear combination translates into a bit-by-bit sum of the component bits of the packets that enter the node.

For example, in **Fig. 4**, each packet p_i, arriving at coding node c has length l, which is $p_i \in \mathbb{F}_2^l$ and each $a_i, b_i \in \mathbb{F}_2$. In node c, the linear combination for each output link is established. During the transmission and propagation process, the new packet $p_1 + p_2$ and length l is forwarded bit-by-bit from node c to each of the adjacent nodes, which are connected to the output links. It is important to highlight that because of this approach, the l-bitwise sums are calculated between the two packets to determine the combined packet.

4. Proposed Problem

4.1 Statement of the Problem

The NC problem (and determining a solution) on a multicast session has been approached in different ways according to [2] [16] [17]. However, in all cases, it has been revised from the contents of the messages or packets, which limits the solution to the size of the field to which the contents belong.

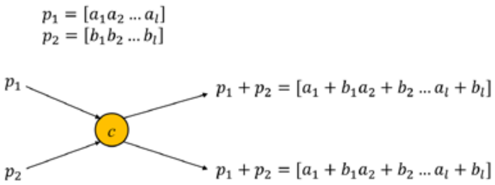

$p_1 = [a_1 a_2 \ldots a_l]$
$p_2 = [b_1 b_2 \ldots b_l]$
p_1
p_2
c
$p_1 + p_2 = [a_1 + b_1 a_2 + b_2 \ldots a_l + b_l]$
$p_1 + p_2 = [a_1 + b_1 a_2 + b_2 \ldots a_l + b_l]$

Fig. 4. The bit-by-bit linear combination in the packets

A multicast session for a communications network is configured with a source node s, a set of sink nodes T, a unit capacity of information per link of length l bits and a maximum flow r for each node in T as shown in **Fig. 5**.

The goal is to determine the tag or code in $\wp$ that names the packets for each of the κ output links $e_1, e_2, \ldots, e_\kappa$ that emerge from the source node s. This process guarantees the reception of independent linear combinations into the sink nodes to retrieve the codes in $\wp_u$ of the original simple r-packets. The packets are sent through the inner links and the nodes of G and are tagged with codes in $\wp$ or specified as linear combinations of the mnemonics in Σ.

This goal includes accommodating (on each outgoing link $e_1, e_2, \ldots, e_\kappa$ from s) the best tag combination (vectors u_i or their linear combinations) in $\wp$ to transfer across the network internally and obtain the tags of the original packets for each node in T and through the solution of a system of linearly independent constraints. That is, the linear combinations generated in r paths that reach each node in T from s must be linearly independent.

The solution is established without considering the size of the field to which the contents of the packets belong. That is, regardless of its length in bits, only the size of the tag or code of each packet generated in the source node will be considered.

In **Fig. 5** and for each outgoing link from s, packets emerge and are generated with either $A_i = g(u_i)$ or linear combinations of codes representing the tags in $\wp_u$. A packet stream must enter each sink node, where the codes representing the stream are in $\wp$ and must be linearly independent. Incoming codes are the result of linear combinations during transmission through the inner nodes.

4.2 Upper Bound of the Number of Outputs

It is important to note that the solution is not unique to any G-graph representing a multicast session. We let σ denote the number of possible combinations of packets sent through the outgoing links from s. The solutions are specified in two terms: for simple packets and for combined packets.

The maximum number of combinations of simple packets emerging from node s through the outgoing links is:

$$\sigma = r^{\kappa}. \tag{8}$$

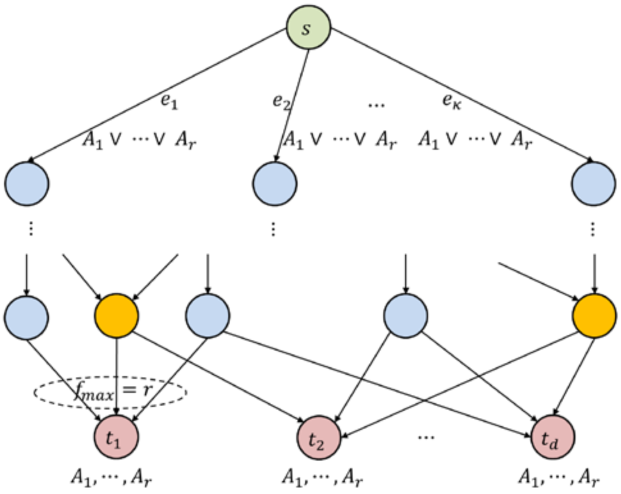


Fig. 5. A general vision of the problem

If it is considered that node (router) s can generate linear combinations from the simple packets entering it, then it would be possible to obtain up to a maximum limit of combinations:

$$\sigma = \left(\sum_{i=1}^{r}\binom{r}{i}\right)^{\kappa} = (2^r - 1)^{\kappa}. \tag{9}$$

It is important to specify that not all combinations are valid solutions to the multicast problem proposed because they would have no solution in the incoming links to the sink nodes.

5. Representation of the Network Model

5.1 Network Adjacency Matrix

An adjacency matrix represents the communication network for a multicast session with the different routes from s to the sinks in $T = \{t_1, t_2, \ldots, t_d\}$. Each cell in the matrix represents the capacity of each of the links that make up the network after obtaining the disjoint paths from s to each sink. This matrix is called C and is $|V| \times |V|$ in size, where:

$$C(i,j) = \begin{cases} 0, & (i,j) \notin E \\ 1, & (i,j) \in E. \end{cases} \tag{10}$$

As previously specified, in link $e = (i,j), i,j \in V, i < j$, a topological ordering of the links in E is reached.

5.2 Classification of the Links

A link $e \in E$ can be classified according to whether $o(e)$ is the source node or not and if $d(e)$ is in T or not, in any of the following types:

$$Type(e) = \begin{cases} 0, & o(e) \neq s, d(e) \notin T \\ 1, & o(e) = s, d(e) \notin T \\ 2, & o(e) \neq s, d(e) \in T \\ 3, & o(e) = s, d(e) \in T. \end{cases} \tag{11}$$

Type 0 corresponds to the inner links, which originate from a node other than s and whose destinations do not belong to set T. Type 1 links are called outgoing links since they originate from s and end in any node that does not belong to T.

Type 2 links are called incoming links since they originate from a node other than s and end in any node belonging to T. Finally, type 3 links are called outgoing-incoming links since they send packets directly from s to a sink in T without passing through inner links.

A limited number of packets arrive at each sink node as determined by the upper boundary of the flow of the multicast network. Since the capacity of each link $e \in E, C(e) = 1$, the number of incoming links that carry the flow to each sink node is equal to

$$r(s,T) \leq min_{t \in T} MinCut(s,t) = min_{t \in T} MaxFlow(s,t). \tag{12}$$

The set of outgoing links is noted as S, which corresponds to the following definition:

$$S = \{e \in E \mid o(e) = s, \ d(e) \notin T\}. \tag{13}$$

The set of incoming links is noted as L and corresponds to the following definition:

$$L = \{e \in E \mid o(e) \neq s, d(e) \in T\}. \tag{14}$$

The set of outgoing-incoming links is noted as K, which is defined as:

$$K = \{e \in E \mid o(e) = s, d(e) \in T\}. \tag{15}$$

From section 4.1, (13), and (15) it was determined that the number of links emerging from s is $|S \cup K| = |S| + |K| = \kappa$; and from (14) and (15) it was established that $|L \cup K| = |L| + |K| = r|T|$.

5.3 Links Table

The links table is constructed as shown in **Table 1** from the adjacency matrix C, which represents G and the classification of the links.

Table 1. Links table format

Index	l
Origin	i
Destination	j
Type	0, 1, 2, and 3

If $C(i,j) = 1$, link $e_l = (i,j)$ with index l contains $Origin(l) = i$ (the same as $o(l) = i$) and $Destination(l) = j$ (the same as $d(l) = j$). $Type(l)$ is specified according to what was previously defined.

Let λ be the number of entries in the links table or the number of links in the graph, which is determined by the number of cells in C whose value is 1 and is defined as:

$$\lambda = \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} C(i,j). \tag{16}$$

6. Solution Bases

6.1 Directed Labeled Line Graph and Hop Matrix

The directed labeled line graph (DLLG) [2] [14] derived from $G = (V, E)$ is $\mathfrak{G} = (\mathcal{V}, \mathcal{E})$, which is constructed from the capacity matrix C with the following rules:

1) Each node $v \in \mathcal{V}$, corresponds to a directed link $e = (i, j) \in E$, then $\mathcal{V} = E$.
2) Each link $e' \in \mathcal{E}$, is defined as $e' = (e_1, e_2)$, where $e_1, e_2 \in E$ and $d(e_1) = o(e_2)$ and therefore $\mathcal{E} = \{(e_1, e_2) \in E^2 | d(e_1) = o(e_2)\}$.
3) Adjacency matrix $F = \left(F(k, l)\right)_{k=1,l=1}^{|\mathcal{V}|}$ is built for graph $\mathfrak{G}$, where any link $e' = (e_1, e_2) \in \mathcal{E}$ is labeled according to (17)

$$F(e_1, e_2) = \begin{cases} 1, & d(e_1) = o(e_2) \\ 0, & otherwise. \end{cases} \tag{17}$$

The matrix of hop F, which results from the above rules, has the following characteristics:

1) It is an upper diagonal matrix with zeros on the main diagonal because it is constructed from a graph whose nodes and links are labeled in topological order.
2) Given the above, it is a nilpotent matrix [26], which indicates $\exists n \in \mathbb{N}$ such that $F^n = 0$.
3) Given the nilpotency of F, the $F^h = \left(F^h(k, l)\right)_{k=1,l=1}^{|\mathcal{V}|}$ matrix is generated, where each $F^h(k, l)$, $1 \le h \le n$, is interpreted according to its value as:

$$F^h(k, l) = \begin{cases} i, & i\ paths\ from\ link\ k\ to\ link\ l\ in\ h\ hops. \\ 0, & no\ path\ from\ link\ k\ to\ link\ l\ in\ h\ hops. \end{cases} \tag{18}$$

6.2 Source-Sinks Hop Matrix

From the matrix of a hop and its properties and using **Algorithm 1**, the Source-Sinks Hop matrix Q is constructed and has a size of $(|S| + |K|) \times |E|$.

Algorithm 1: Construction of the Source-Sinks Hop matrix

Input: $\boldsymbol{F}, \boldsymbol{S} \cup \boldsymbol{K}, \boldsymbol{E}$

Output: Source-Sinks Hop matrix $\boldsymbol{Q}$

```
//The rows of $Q$ are constituted by links $k$ such that $k \in S \cup K$.
//The columns of $Q$ are constituted by every link $l \in E$ (including the inner or type 0 links, which are not necessary for the final model).
Initialize $Q$ with $Q(k, l) = 0, (\forall k \in S \cup K, (\forall l \in E))$
For each $F^h, (1 \le h < n)$ //Actualization of $Q$
  For each $k \in S \cup K$
    For each $l \in E$
      If $F^h(k, l) >= 1$ and $Type(l) = 2$
        $Q(k, l) += F^h(k, l)$ //Increases by $F^h(k, l)$ each time
      End If
    End For
  End For
End For
Return $Q$
```

In conclusion, each $Q(k, l) \neq 0$ represents the number of times that the outgoing link flow k ($Type(k) = 1$), contributes in a linear combination to the calculation of the incoming link flow l ($Type(l) = 2$). That is, if f_k corresponds to the outgoing flow from s and is transmitted through the outgoing link k, then $Q(k,l) * f_k$ is the flow that will transit in a linear combination through the incoming link l to the sink node $d(l)$. Thus, f_k will contribute $Q(k,l)$ times to the linear combination of the incoming flow through the incoming link l to the sink node $d(l)$. Actually, the flow from which the outgoing link k will contribute is f_k if $Q(k,l)$ is an odd integer and it is zero, otherwise.

Let $\mathcal{L} = \{l \in E \mid d(l) \notin T\}$ and the links with destinations not in T, i.e., $Type(l) = 0$ or $Type(l) = 1$. From this definition and previous definitions, the following theorems are established:

Theorem 1: Let $\boldsymbol{l \in L}$, then $\boldsymbol{Q^T(l) \neq 0}$ ($\boldsymbol{Q^T(l)}$ denotes the $\boldsymbol{l^{th}}$ column of matrix $\boldsymbol{Q}$ and $\boldsymbol{Q^T}$ is the transpose of $\boldsymbol{Q}$).

Proof: If $l \in L$, by definition of set L, then $d(l) \in T$. Besides, $Type(l) = 2$ and so for each iteration of Algorithm 1 until $F^n = 0$ is reached, when there are h hops from k to l; that is, when $F^h(k,l) = i$, the increase of cell $Q(k,l)$ is achieved, and this prevents $Q^T(l) = 0$ ■

Theorem 2: Let $\boldsymbol{l \in \mathcal{L}}$, then $\boldsymbol{Q^T(l) = 0}$.

Proof: If $l \in \mathcal{L}$, then by definition of set $\mathcal{L}$, $d(l) \notin T$. Besides, $Type(l) \neq 2$ and $Type(l) \neq 3$, and so for each iteration of **Algorithm 1** until $F^n = 0$ is reached, the increase in cell $Q(k,l)$ will not be achieved and this will lead to $Q^T(l) = 0$ ■

Theorem 3: If $\boldsymbol{l}$ is a link where $\boldsymbol{Type(l) = 3}$, then $\boldsymbol{Q^T(l) = 0}$.

Proof: If l is a type 3 link, then $l \in K$ and by definition $o(l) = s$ and $d(l) = t \in T$. This indicates that l is a direct link from s to t with no intermediate nodes. Thus, there are no links before or after l, and consequently, l is an incoming link with no hops where each iteration of **Algorithm 1** produces $F^h(k,l) = 0, \forall k \in S \cup K$ and results in $Q^T(l) = 0$ ■

Q contains $|\mathcal{L}|$ columns in 0, and these are called non-sink columns, which do not show any relationship between an outgoing link and an incoming link or specifically between node s and some $t \in T$. Furthermore, they fulfill **Theorem 2**.

The rest will be called sink columns, which are constituted by columns l whose $d(l) \in T$, i.e., the columns that fulfill **Theorem 1** or **Theorem 3**.

The running time of **Algorithm 1** is $O(n|S \cup K||E|) = O(n\kappa|E|)$.

6.3 Reduced Source-Sinks Hop Matrix

In the Source-Sinks Hop matrix, the $|\mathcal{L}|$ columns in 0 (non-sink columns) that correspond with links $l' \in E$ such that $d(l') \notin T$, are eliminated to get the Reduced Source-Sinks Hop matrix, i.e., only the columns corresponding to type 2 or type 3 links are kept. Q_r denotes the Reduced Source-Sinks Hop matrix, and $(|S| + |K|) \times (|L| + |K|) = \kappa r|T|$ is its dimension. **Algorithm 2** shows the process of reducing the Source-Sinks Hop matrix.

```
Algorithm 2: Reduced Source-Sinks Hop Matrix
Input: $Q, L \cup K, S \cup K$
Output: $Q_r$
Initialize $Q_r$ with $Q_r(k,l) = 0, (\forall k \in S \cup K, (\forall l \in L \cup K))$
For each link $l \in L \cup K$
    $Q_r^T(l) = Q^T(l)$
End For
Return $Q_r$
```

The running time of **Algorithm 2** is $O(|S \cup K||L \cup K|) = O(\kappa r|T|)$.

7. Construction of constraints

7.1 Principles of the Proposed Method

The proposed model starts with the graph $G = (V, E)$, which represents a multicast network of a single source node s, a set of sink nodes T, maximum flow r, and capacity $C(e) = 1$ for each link $e \in E$. The goal was to construct a system of linear constraints formed by variables that represent the outgoing, incoming, and outgoing-incoming packet flows, where each flow is represented by the code associated with the packet (a simple or a combined packet) that can travel through the link. The model is based on three principles:

7.1.1 Definition of Constraints in the Source Node

In node s, different packets are created and distinguished with a tag length of r-bits and with $A_i = g(u_i)$, where u_i is the vector defined in section 3.4. From this node, $|S| + |K|$ links (outgoing and outgoing-incoming) emerge and allow the establishment of r possible output packets for each of these links. Then, it is deduced that the following simple flows (a single packet per outgoing or outgoing-incoming link) can be obtained according to the relationship:

$$f_e = A_1 |\, A_2 |\, \ldots |A_r, \forall e \in S \cup K. \tag{19}$$

It is possible that for each link $\in S \cup K$, there is a linear combination (combined packets) of simple packets emerging, tagged with some u_i, and generating a packet p_e tagged $c(p_e) \in \wp$ that is stored in the combined flow f_e:

$$f_e = \sum_{u_i \in \wp} u_i = \sum_{u_i \in \wp, g(u_i) = A_i} A_i \,, \forall e \in S. \tag{20}$$

The previous equation can also be written as:

$$f_e \neq 0. \tag{21}$$

for every case where $f_e \in \mathbb{F}_2^r \setminus 0$ o $f_e \in \wp$.

7.1.2 Definition of the Constraints on Hops from an Outgoing to an Incoming Link

Given the graph of **Fig. 6**, let f_{sv_1} be a flow that emerges from s, which will arrive through the different links that form the path, and by the principle of the conservation of flow [21] [27], the following relationship of equality is deduced:

$$f_{sv_1} = f_{v_1 v_2} = \cdots = f_{v_{j-1} v_j} = f_{v_j t}. \tag{22}$$

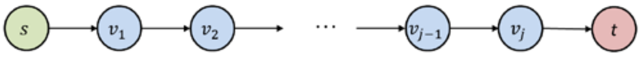


Fig. 6. The path from a source node to a sink node

This equality should not be taken in a strictly literal sense. Equality, in this case, indicates that the flow to the left of equality contributes to the flow of the right. This means that it will be part of the linear combination built to obtain the flow through each output link at each intermediate node of the path. That is, the flow of one output link is increased by the flow preceding it within this equality.

On a path from s to sink t, let f_k and f_l be flows such that $f_k = f_l$ through the links k and l, where $d(k) = o(l)$. From this equality two scenarios can be derived:

(a) If $d(k)$ is a classical routing node, the incoming flow f_k is the same outgoing flow f_l, since there is no linear coding at the output and the flow enters through the input link and passes identically to the output link.

(b) If $d(k)$ is a coding node, the incoming flow f_k is a component of the linear combination formed in the output links of the node, that is:

$$f_l = \cdots + f_k + \cdots. \tag{23}$$

From the scenario in (a) and (22), if there is no coding node in the path from s to t, then:

$$f_{sv_1} = f_{v_jt}. \tag{24}$$

That is, the flow originates at source node s, which is transmitted through the outgoing link (s, v_1) and is the same in the incoming link (v_j, t) after j hops.

If there is a coding node in graph G, paths will be generated from s to sink t as shown in **Fig. 7**, where two outgoing paths from s converge on the coding node v_{j-1}. Hence, the following two relationships of the equality of flows from s to t are established:

$$f_{sv_1} = f_{v_1v_2} = \cdots = f_{v_{j-1}v_j} = f_{v_jt}. \tag{25}$$

and

$$f_{sw_1} = f_{w_1w_2} = \cdots = f_{v_{j-1}v_j} = f_{v_jt}. \tag{26}$$

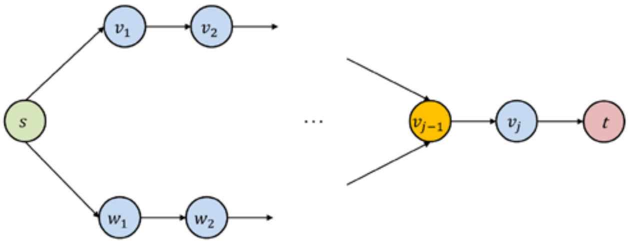


Fig. 7. The confluence of paths at the coding node

It is observed that (v_{j-1}, v_j) is a bottleneck link and flow $f_{v_{j-1}v_j}$ originates in coding node v_{j-1}. Also, flows $f_{sv_1}$and f_{sw_1} emerge from s through links (s, v_1) and (s, w_1), respectively. According to (25) and (26), they contribute to $f_{v_{j-1}v_j}$, and according to (23), they form the linear combination:

$$f_{v_{j-1}v_j} = f_{sv_1} + f_{sw_1}. \tag{27}$$

Therefore, from (25), (26) and (27), it is deduced that the flow in the incoming link (v_j, t) will directly depend on the outgoing flows from s through the outgoing links (s, v_1) and (s, w_1) by the relation:

$$f_{v_jt} = f_{sv_1} + f_{sw_1}. \tag{28}$$

In both (25) and (28), the flows in the incoming links to the node $t \in T$ correspond to packets of size l-bits over the vector space $\mathbb{F}_2^l$ as specified in section 3.3, which is the same as the packets of the outgoing flows.

7.1.3 Definition of Constraints in the Incoming Links

In every sink node, $t \in T$, the following rules must be observed:

(a) Incoming flows must be non-zero; if f_{v_jt} is the flow arriving at t and through the incoming link (v_j, t), the packet flowing through this link and incoming to t must have a tag or linear combination that is different to 0; that is, according to **Fig. 8**, it must be observed that:

$$f_{v_jt} \neq 0, \forall\left(v_j, t\right) \in L. \tag{29}$$

(b) The linear combination of the incoming flows to a sink node must be non-zero; similarly, it must be observed that the linear combination of the packet codes (or tags) that enter each sink node through the incoming flows must be a value other than 0. That is, according to **Fig. 8**, the following constraint is established:

$$\sum_{j=1}^{r} f_{v_jt} \neq 0, \forall t \in T. \tag{30}$$

(c) The matrix formed from codes that tag the incoming link flows must have a range r; the packet codes that arrive within each flow to the node t must be linearly independent. That is, the bits that make up the code of each packet in each incoming flow to sink node t form the rows of square matrix P of dimension $r \times r$ with $range(P) = r$. The rows of P represent the codes of the r-packets with size r-bits that enter the sink node.

(d)

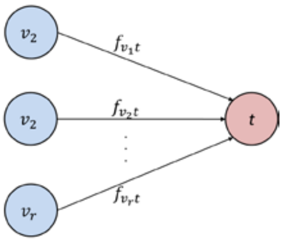


Fig. 8. Incoming links

The shape of the matrix P, which is constituted by the r-codes that identify the packets of the flows that arrive at a sink node in T, is:

$$P = \begin{bmatrix} c(p_1) \\ c(p_2) \\ \vdots \\ c(p_r) \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1r} \\ b_{12} & b_{22} & \dots & b_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ b_{r1} & b_{r2} & \dots & b_{rr} \end{bmatrix}. \tag{31}$$

For example, if $r = 3$, this would indicate a maximum flow for a multicast session of 3 packets generated at the source node and reaching each sink node. The length of the packets is indifferent; therefore, the codes are over the vector space $\mathbb{F}_2^3$. If it is assumed that a sink node receives the packets tagged with mnemonics $A, A + C$ and $B + C$ on its three links, then they correspond to the following binary codes:

$$\begin{aligned} A \quad &= 001 = c(p_1) \\ A + C &= 101 = c(p_2) \\ B + C &= 110 = c(p_3) \end{aligned}$$

The corresponding matrix $P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ and its determinant is 1.

7.2 Creation Rules of the System of Constraints

7.2.1 Constraints on Outgoing Link Flows

For each link e of type 1 or 3, the variable names of the outgoing link flow (packet) codes are defined according to the following rule: if e is an outgoing link in $S \cup K$ and its $o(e) = s$, then the notation of the outgoing link flow variable is $f_{s\_d(e)}$, and its computational size is

r-bits. The solutions for the outgoing flow $f_{s\_d(e)}$ are classified according to:

(a) Simple packets: each $f_{s\_d(e)}$ corresponds to an integer power of 2 from 1 to 2^r:

$$f_{s\_d(e)} = 1 \vee 2 \vee 4 \vee \ldots \vee 2^r, \text{where} \vee \text{ is logical } or \text{ operator.} \tag{32}$$

(b) Combined packets: each $f_{s\_d(e)}$ corresponds to an integer between 1 and 2^r:

$$f_{s\_d(e)} = [1, 2^r] \neq 0. \tag{33}$$

7.2.2 Constraints on incoming link flows

For each link l of type 2 or 3, the variable names of the incoming link flow (packet) codes are defined according to the following rule: if l is a link in $L \cup K$ and its $d(l) = t, t \in T$, then the notation of the incoming link flow code variable is $f_{o(l)\_t}$, and its computational size is r-bits.

The constraints are set with the support of the Reduced Source-Sinks Hop matrix Q_r, where each $Q_r(k, l) = j$ indicates that there is a direct path with j hops ($1 \leq j \leq n-1$ and $F^n = 0$) between the outgoing link k and the incoming link l. $Q_r^T(l)$ is the column corresponding to the incoming link l of the matrix Q_r, and $Q_r^T(l, k)$ is the number of times that the flow coming from the outgoing link k (noted as $f_{s\_d(k)}$) will contribute to the linear combination built to obtain the flow of the incoming link l (noted as $f_{o(l)\_t}$). For an incoming link l, the flow $f_{o(l)\_t}$ is written as a linear combination of the following number of added components:

$$\eta_{f_{o(l)\_t}} = \sum_{k \in S \cup K} Q_r^T(l, k). \tag{34}$$

From (28), the flow $f_{o(l)\_t}$, in the incoming link l, is obtained through the following expression and formed by the added outgoing flows $f_{s\_d(k)}, \forall k \in S \cup K$:

$$f_{o(l)\_t} = \sum_{k \in S \cup K} \sum_{i=1}^{Q_r^T(l,k)} f_{s\_d(k)} \,. \tag{35}$$

7.2.3 Constraints on incoming links and sink nodes

The constraints given in section 7.1.3 must be met.

(a) For each link l of type 2 or 3 and considering (29), the flow must fulfill the expression:

$$f_{o(l)\_t} \neq 0, \forall t \in T, \forall l \text{ with } d(l) = t. \tag{36}$$

(b) For the linear combination of the flows in the incoming links $l_1, l_2, \ldots, l_r$ to the sink node t and considering (30), the following expression must be satisfied:

$$\sum_{i=1}^{r} f_{o(l_i)\_t} \neq 0, \forall t \in T, \forall l_i \text{ with } d(l_i) = t. \tag{37}$$

(c) To establish the linear independence between the codes of flows (packets) arriving at a sink node $t \in T$, a matrix P_t is constructed whose rows contain these codes. Each of the r-bits that make up a flow $f_{o(l_i)\_t}$, will be denoted by $f_{o(l_i)\_t}^{(j)}$, where $1 \leq i, j \leq r$:

$$P_t = \underbrace{\begin{bmatrix} f_{o(l_1)\_t} \\ f_{o(l_2)\_t} \\ \vdots \\ f_{o(l_r)\_t} \end{bmatrix}}_{r-bits} = \begin{bmatrix} f_{o(l_1)\_t}^{(1)} & f_{o(l_1)\_t}^{(2)} & \cdots & f_{o(l_1)\_t}^{(r)} \\ f_{o(l_2)\_t}^{(1)} & f_{o(l_2)\_t}^{(2)} & \cdots & f_{o(l_2)\_t}^{(r)} \\ \vdots & \vdots & \ddots & \vdots \\ f_{o(l_r)\_t}^{(1)} & f_{o(l_r)\_t}^{(2)} & \cdots & f_{o(l_r)\_t}^{(r)} \end{bmatrix}. \tag{38}$$

The determinant of P_t will be calculated by the method of minors and cofactors in the following manner for sink node t:

$$f_{o(l_1)\_t}^{(1)} * M_{11} + f_{o(l_1)_t}^{(2)} * M_{12} + \cdots + f_{o(l_1)_t}^{(r)} * M_{1r} \neq 0. \tag{39}$$

This determinant must be a non-zero value to fulfill the linear independence, that is:

$$det(P_t) = \sum_{i=1}^{r} f^{(i)}_{o(l_1)_t} * M_{1r} \neq 0, \forall t \in T. \tag{40}$$

In short, a system of constraints is achieved and if solved will determine the value of the code that tags the packet emerging by each outgoing link that is born in the source node s. Similarly, the values of the codes that tag the packets arriving at each sink node in T will be obtained. The system of constraints used to determine the solution to the multicast session of a single source node s and the set of sink nodes T is summarized as:

$$f_{s\_d(k)} = 1 \vee 2 \vee 4 \vee \ldots \vee 2^r \quad \text{or } f_{s\_d(k)} = [1:2^r] \neq 0.$$

$$f_{o(l)\_t} = \sum_{k \in S \cup K} \sum_{i=1}^{Q_r^T(l,k)} f_{s\_d(k)} \cdot$$

$$f_{o(l)\_t} \neq 0, \forall t \in T, \forall l \text{ with } d(l) = t.$$

$$\sum_{i=1}^{r} f_{o(l_i)\_t} \neq 0\,, \forall t \in T, \forall l_i \text{ with } d(l_i) = t.$$

$$P_t = \underbrace{\begin{bmatrix} f_{o(l_1)\_t} \\ f_{o(l_2)\_t} \\ \vdots \\ f_{o(l_r)\_t} \end{bmatrix}}_{r-bits} = \begin{bmatrix} f^{(1)}_{o(l_1)\_t} & f^{(2)}_{o(l_1)\_t} & \cdots & f^{(r)}_{o(l_1)\_t} \\ f^{(1)}_{o(l_2)\_t} & f^{(2)}_{o(l_2)\_t} & \cdots & f^{(r)}_{o(l_2)\_t} \\ \vdots & \vdots & \ddots & \vdots \\ f^{(1)}_{o(l_r)\_t} & f^{(2)}_{o(l_r)\_t} & \cdots & f^{(r)}_{o(l_r)\_t} \end{bmatrix}.$$

$$\det(P_t) = \sum_{i=1}^{r} f^{(i)}_{o(l_1)\_t} * M_{1r} \neq 0, \forall t \in T.$$

8. Examples and Results

8.1 Example 1: Multicast session with nine nodes

8.1.1 System graphs and matrices

Fig. 9(a) shows the graph of a multicast session with nine nodes and with a source node ($s = 1$), three sinks ($T = \{7,8,9\}$) and maximum flow of $r = 2$ packets in $\mathbb{F}_2^l$. With a maximum flow of 2, the codes that will tag the packets are nonzero elements of the vector space $\mathbb{F}_2^2$. In addition, there is a coding node (node 5) that is not relevant to the model's approach. **Table 2** shows the links and their types. From this table, it is also inferred that the sets of outgoing and incoming links are constituted by:
$S = \{(1,2),(1,3),(1,4)\}$ and $L = \{(2,7),(6,7),(3,9),(6,8),(4,8),(6,9)\}$.

Fig. 9(b) shows the DLLG for the multicast session with nine nodes. The longest path is three hops, and so it can only be iterated to $n = 3$ and with $n = 4$, $F^4 = 0$. After obtaining the intermediate matrices, the Source-Sinks Hop matrix and the Reduced Source-Sinks Hop matrix are deduced as shown in **Table 3** and **Table 4**, respectively.

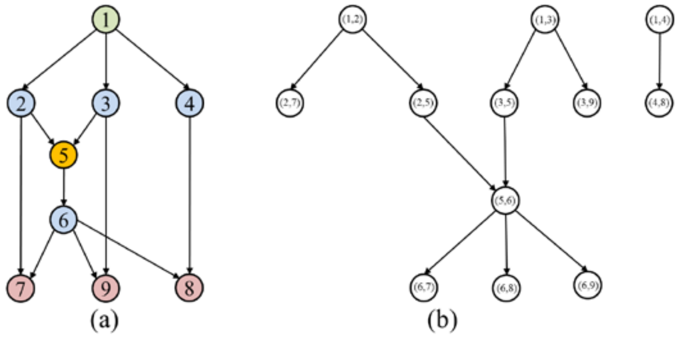


Fig. 9(a) Multicast session for G with nine nodes. **(b)** Directed Labeled Line Graph for G with nine nodes

Table 2. Links table

Index	1	2	3	4	5	6	7	8	9	10	11	12
Origin	1	1	1	2	2	3	3	4	5	6	6	6
Destination	2	3	4	5	7	5	9	8	6	7	8	9
Type	1	1	1	0	2	0	2	2	0	2	2	2

Table 3. Source-Sinks Hop Matrix Q

Link	**(1,2)**	**(1,3)**	**(1,4)**	**(2,5)**	**(2,7)**	**(3,5)**	**(3,9)**	**(4,8)**	**(5,6)**	**(6,7)**	**(6,8)**	**(6,9)**
(1,2)	0	0	0	0	**1**	0	0	0	0	**1**	**1**	**1**
(1,3)	0	0	0	0	0	0	**1**	0	0	**1**	**1**	**1**
(1,4)	0	0	0	0	0	0	0	**1**	0	0	0	0

Table 4. Reduced Source-Sinks Hop Matrix Q_r

Link	**(2,7)**	**(3,9)**	**(4,8)**	**(6,7)**	**(6,8)**	**(6,9)**
(1,2)	**1**	0	0	**1**	**1**	**1**
(1,3)	0	**1**	0	**1**	**1**	**1**
(1,4)	0	0	**1**	0	0	0

8.1.2 System constraints

According to the multicast session graph represented in **Fig. 9**, the following set of constraints is derived from applying the model.

(a) Constraints on the outgoing link flows:

Case I: Flows with simple packets

$f_{1\_2} = 1V2; f_{1\_3} = 1V2; f_{1\_4} = 1V2$

Case II: Flows with combined packets

$f_{1\_2} \neq 0; f_{1\_3} \neq 0; f_{1\_4} \neq 0$

(b) Constraints on the incoming link flows:

With the support of the Reduced Source-Sinks Hop matrix Q_r, it is determined that:

$f_{2\_7} = f_{1\_2}$; $f_{3\_9} = f_{1\_3}$; $f_{4\_8} = f_{1\_4}$; $f_{6\_7} = f_{1\_2} + f_{1\_3}$; $f_{6\_8} = f_{1\_2} + f_{1\_3}$; $f_{6\_9} = f_{1\_2} + f_{1\_3}$

(c) Constraints on the incoming links and the sink nodes:

(1) Non-zero incoming flows:

$f_{2\_7} \neq 0; f_{3\_9} \neq 0; f_{4\_8} \neq 0; f_{6\_7} \neq 0; f_{6\_8} \neq 0; f_{6\_9} \neq 0$

(2) Non-zero linear combination of incoming flows to a sink node:

$$f_{2\_7} + f_{6\_7} \neq 0$$
$$f_{4\_8} + f_{6\_8} \neq 0$$
$$f_{3\_9} + f_{6\_9} \neq 0$$

(3) Matrix of tags sized $r = 2$:
The codes of this field are:

$$\wp = \mathbb{F}_2^2 \backslash \{0\} = \{01{,}10{,}11\}$$

Next, the code matrix and the independent linearity constraint between the incoming flows for each sink node T are shown in **Table 5**.

Table 5. Code Matrix and Independent Linearity Constraints

For $t = 7$	**For $t = 8$**	**For $t = 9$**
$P_7 = \begin{bmatrix} f_{2\_7}^1 & f_{2\_7}^2 \\ f_{6\_7}^1 & f_{6\_7}^2 \end{bmatrix}$	$P_8 = \begin{bmatrix} f_{4_8}^1 & f_{4_8}^2 \\ f_{6_8}^1 & f_{6_8}^2 \end{bmatrix}$	$P_9 = \begin{bmatrix} f_{3\_9}^1 & f_{3\_9}^2 \\ f_{6\_9}^1 & f_{6\_9}^2 \end{bmatrix}$
$f_{2\_7}^1 * f_{6\_7}^2 + f_{2\_7}^2 * f_{6\_7}^1 \neq 0$	$f_{4\_8}^1 * f_{6\_8}^2 + f_{4\_8}^2 * f_{6\_8}^1 \neq 0$	$f_{3\_9}^1 * f_{6\_9}^2 + f_{3\_9}^2 * f_{6\_9}^1 \neq 0$

8.1.3 System Solution

When carrying this system to an equation solver (Z3[28] software, which is used as a theorem tester and first-order logic solver), the following results are obtained for each flow value (packet) in the outgoing links and each flow value (packet) in the incoming links:

Case I: Flows with simple packets:

Outgoing packet codes per outgoing link:

$$f_{1\_2} = 2 = B$$
$$f_{1\_3} = 1 = A$$
$$f_{1\_4} = 2 = B$$

The incoming packet codes per incoming link are shown in **Table 6**.

Table 6. Incoming Packet Codes per Incoming Link- Case I

For $t = 7$	**For $t = 8$**	**For $t = 9$**
$f_{2\_7} = 2 = B$ $f_{6\_7} = 3 = A + B$	$f_{4\_8} = 2 = B$ $f_{6\_8} = 3 = A + B$	$f_{3\_9} = 1 = A$ $f_{6\_9} = 3 = A + B$

The multicast session graph with the solution is shown in **Fig. 10(a)**.

Case II: Flows with combined packets:

Outgoing packet codes per outgoing link:

$$f_{1\_2} = 1 = A$$
$$f_{1\_3} = 3 = A + B$$
$$f_{1\_4} = 3 = A + B$$

The incoming packet codes per incoming link are shown in **Table 7**.

Table 7. Incoming Packet Codes per Incoming Link – Case II

For $t = 7$	**For $t = 8$**	**For $t = 9$**
$f_{2\_7} = 1 = A$ $f_{6\_7} = 2 = B$	$f_{4\_8} = 3 = A + B$ $f_{6\_8} = 2 = B$	$f_{3\_9} = 3 = A + B$ $f_{6\_9} = 2 = B$

The multicast session graph with the solution is shown in **Fig. 10(b)**.

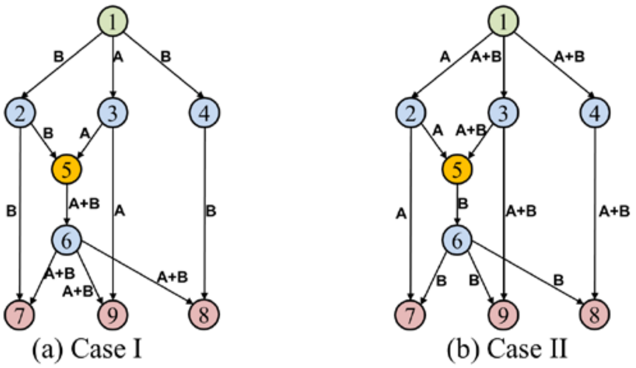


Fig. 10. Solutions in graph G with nine nodes.

8.2 Example 2: Multicast session with 18 nodes

8.2.1 System graphs and matrices

Fig. 11(a) shows the graph of a multicast session with 18 nodes and with a source node ($s = 1$), three sinks ($T = \{18,19,20\}$) and maximum flow of $r = 4$ packets in $\mathbb{F}_2^l$. With a maximum flow of 4, the codes that will tag the packets are nonzero elements of the vector space $\mathbb{F}_2^4$. Also, there are coding nodes (nodes 8 and 12) that are not relevant to the model's approach. **Table 8** shows the links and their types. From this table, it is also inferred that the sets of outgoing and incoming links are constituted by:

$$S = \{(1,2),(1,3),(1,4),(1,5),(1,6),(1,7)\}$$

$$L = \left\{\begin{matrix}(13,18),(14,18),(15,18),(16,18),(6,19),(14,19),(15,19),(16,19),(10,20), \\ (14,20),(15,20),(16,20)\end{matrix}\right\}$$

Fig. 11(b) shows the DLLG for the multicast session with 18 nodes. The longest path is three hops, and so it can only be iterated to $n = 3$ and with $n = 4$, $F^4 = 0$. After obtaining the intermediate matrices, the Source-Sinks Hop matrix and the Reduced Source-Sinks Hop matrix are deduced as shown in **Table 9** and **Table 10**, respectively.

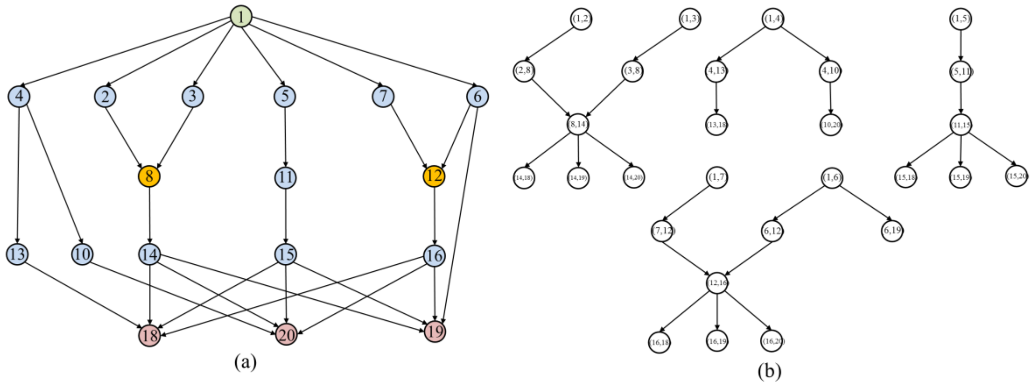


Fig. 11(a) Multicast session for G with 18 nodes. **(b)** Directed Labeled Line Graph for G with 18 nodes.

Table 8. Links table

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Origin	1	1	1	1	1	1	2	3	4	4	5	6	6	7
Destination	2	3	4	5	6	7	8	8	10	13	11	12	19	12
Type	1	1	1	1	1	1	0	0	0	0	0	0	2	0

Index	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Origin	8	10	11	12	13	14	14	14	15	15	15	16	16	16
Destination	14	20	15	16	18	18	19	20	18	19	20	18	19	20
Type	0	2	0	0	2	2	2	2	2	2	2	2	2	2

8.2.2 System constraints

According to the multicast session graph represented in **Fig. 11**, the following set of constraints is derived from applying the model.

(a) Constraints on the outgoing link flows:

Case I: Flows with simple packets

$f_{1\_2} = 1V2V4V8$; $f_{1\_3} = 1V2V4V8$; $f_{1\_4} = 1V2V4V8$;
$f_{1\_5} = 1V2V4V8$; $f_{1\_6} = 1V2V4V8$; $f_{1\_7} = 1V2V4V8$

Case II: Flows with combined packets

$f_{1\_2} \neq 0$; $f_{1\_3} \neq 0$; $f_{1\_4} \neq 0$; $f_{1\_5} \neq 0$; $f_{1\_6} \neq 0$; $f_{1\_7} \neq 0$

(b) Constraints on the incoming link flows:

With the support of the Reduced Source-Sinks Hop matrix Q_r, it is determined that:

$f_{13\_18} = f_{1\_4}$; $f_{14\_18} = f_{1\_2} + f_{1\_3}$; $f_{15\_18} = f_{1\_5}$; $f_{16\_18} = f_{1\_6} + f_{1\_7}$; $f_{6\_19} = f_{1\_6}$;
$f_{14\_19} = f_{1\_2} + f_{1\_3}$; $f_{15\_19} = f_{1\_5}$; $f_{16\_19} = f_{1\_6} + f_{1\_7}$; $f_{10\_20} = f_{1\_4}$; $f_{14\_20} = f_{1\_2} + f_{1\_3}$;
$f_{15\_20} = f_{1\_5}$; $f_{16\_20} = f_{1\_6} + f_{1\_7}$

Table 9. Source-Sinks Hop Matrix Q

Link	**(1,2)**	**(1,3)**	**(1,4)**	**(1,5)**	**(1,6)**	**(1,7)**	**(2,8)**	**(3,8)**	**(4,10)**	**(4,13)**	**(5,11)**	**(6,12)**	**(6,19)**	**(7,12)**	**(8,14)**	**(10,20)**	**(11,15)**	**(12,16)**	**(13,18)**	**(14,18)**	**(14,19)**	**(14,20)**	**(15,18)**	**(15,19)**	**(15,20)**	**(16,18)**	**(16,19)**	**(16,20)**
(1,2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	0	0	0	0	0	0
(1,3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	0	0	0	0	0	0
(1,4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	**1**	0	0	0	0	0	0	0	0	0
(1,5)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	0	0	0
(1,6)	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**
(1,7)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**

Table 10. Reduced Source-Sinks Hop Matrix Q_r

Link	**(6,19)**	**(10,20)**	**(13,18)**	**(14,18)**	**(14,19)**	**(14,20)**	**(15,18)**	**(15,19)**	**(15,20)**	**(16,18)**	**(16,19)**	**(16,20)**
(1,2)	0	0	0	**1**	**1**	**1**	0	0	0	0	0	0
(1,3)	0	0	0	**1**	**1**	**1**	0	0	0	0	0	0
(1,4)	0	**1**	**1**	0	0	0	0	0	0	0	0	0
(1,5)	0	0	0	0	0	0	**1**	**1**	**1**	0	0	0
(1,6)	**1**	0	0	0	0	0	0	0	0	**1**	**1**	**1**
(1,7)	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**

(c) Constraints on the incoming links and the sink nodes:

(1) Non-zero incoming flows:

$f_{13\_18} \neq 0$; $f_{14\_18} \neq 0$; $f_{15\_18} \neq 0$; $f_{16\_18} \neq 0$; $f_{6\_19} \neq 0$; $f_{14\_19} \neq 0$; $f_{15\_19} \neq 0$; $f_{16\_19} \neq 0$; $f_{10\_20} \neq 0$; $f_{14\_20} \neq 0$; $f_{15\_20} \neq 0$; $f_{16\_20} \neq 0$

(2) Non-zero linear combination of incoming flows to a sink node:

$$f_{13\_18} + f_{14\_18} + f_{15\_18} + f_{16\_18} \neq 0$$
$$f_{6\_19} + f_{14\_19} + f_{15\_19} + f_{16\_19} \neq 0$$
$$f_{10\_20} + f_{14\_20} + f_{15\_20} + f_{16\_20} \neq 0$$

(3) Matrix of tags sized $r = 4$:

The codes of this field are:

$\wp = \mathbb{F}_2^4 \backslash \{0\} = \{0001,0010,0011,0100,0101,0110,0111,1000,1001,1010,1011,1100,1101,1110,1111\}$

Next, the code matrix and the independent linearity constraint between the incoming flows for each sink node T are shown in **Table 11**.

Table 11. Code Matrix and Independent Linearity Constraints

For t = 18	**For t = 19**	**For t = 20**
$P_{18} = \begin{bmatrix} f^1_{13\_18} & f^2_{13\_18} & f^3_{13\_18} & f^4_{13\_18} \\ f^1_{14\_18} & f^2_{14\_18} & f^3_{14\_18} & f^4_{14\_18} \\ f^1_{15\_18} & f^2_{15\_18} & f^3_{15\_18} & f^4_{15\_18} \\ f^1_{16\_18} & f^2_{16\_18} & f^3_{16\_18} & f^4_{16\_18} \end{bmatrix}$	$P_{19} = \begin{bmatrix} f^1_{6\_19} & f^2_{6\_19} & f^3_{6\_19} & f^4_{6\_19} \\ f^1_{14\_19} & f^2_{14\_19} & f^3_{14\_19} & f^4_{14\_19} \\ f^1_{15\_19} & f^2_{15\_19} & f^3_{15\_19} & f^4_{15\_19} \\ f^1_{16\_19} & f^2_{16\_19} & f^3_{16\_19} & f^4_{16\_19} \end{bmatrix}$	$P_{20} = \begin{bmatrix} f^1_{10\_20} & f^2_{10\_20} & f^3_{10\_20} & f^4_{10\_20} \\ f^1_{14\_20} & f^2_{14\_20} & f^3_{14\_20} & f^4_{14\_20} \\ f^1_{15\_20} & f^2_{15\_20} & f^3_{15\_20} & f^4_{15\_20} \\ f^1_{16\_20} & f^2_{16\_20} & f^3_{16\_20} & f^4_{16\_20} \end{bmatrix}$
$\det(P_{18}) \neq 0$	$\det(P_{19}) \neq 0$	$\det(P_{20}) \neq 0$

8.2.3 System Solution

The following results are obtained for each flow value (packet) in the outgoing and incoming links:

Case I: Flows with simple packets:

Outgoing packet codes per outgoing link:

$f_{1\_2} = 1 = A$ $\quad f_{1\_4} = 1 = A$ $\quad f_{1\_6} = 2 = B$

$f_{1\_3} = 8 = D$ $\quad f_{1\_5} = 4 = C$ $\quad f_{1\_7} = 1 = A$

The incoming packet codes per incoming link are shown in **Table 12**.

Table 12. Incoming Packet Codes per Incoming Link- Case I

For t = 18	**For t = 19**	**For t = 20**
$f_{13\_18} = 1 = A$	$f_{6\_19} = 2 = B$	$f_{10\_20} = 1 = A$
$f_{14\_18} = 9 = A + D$	$f_{14\_19} = 9 = A + D$	$f_{14\_20} = 9 = A + D$
$f_{15\_18} = 4 = C$	$f_{15\_19} = 4 = C$	$f_{15\_20} = 4 = C$
$f_{16\_18} = 3 = A + B$	$f_{16\_19} = 3 = A + B$	$f_{16\_20} = 3 = A + B$

The multicast session graph with the solution is shown in **Fig. 12(a)**.

Case II: Flows with combined packets:

Outgoing packet codes per outgoing link:

$f_{1\_2} = 7 = A + B + C$ $\quad f_{1\_4} = 3 = A + B$ $\quad f_{1\_6} = 3 = A + B$

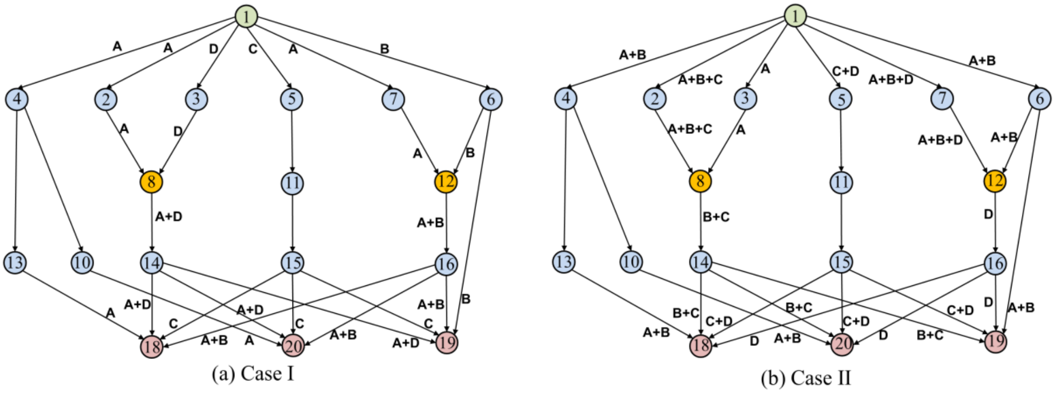
$f_{1\_3} = 1 = A$ $\quad f_{1\_5} = 12 = C + D$ $\quad f_{1\_7} = 11 = A + B + D$

The incoming packet codes per incoming link are shown in **Table 13**.

Table 13. Incoming Packet Codes per Incoming Link – Case II

For t = 18	**For t = 19**	**For t = 20**
$f_{13\_18} = 3 = A + B$	$f_{6\_19} = 3 = A + B$	$f_{10\_20} = 3 = A + B$
$f_{14\_18} = 6 = B + C$	$f_{14\_19} = 6 = B + C$	$f_{14\_20} = 6 = B + C$
$f_{15\_18} = 12 = C + D$	$f_{15\_19} = 12 = C + D$	$f_{15\_20} = 12 = C + D$
$f_{16\_18} = 8 = D$	$f_{16\_19} = 8 = D$	$f_{16\_20} = 8 = D$

The multicast session graph with the solution is shown in **Fig. 12(b)**.



(a) Case I

(b) Case II

Fig. 12. The solutions in graph G with 18 nodes.

5. Conclusion

Finding a solution to the multicast NC problem is one of the challenges addressed in several previous studies [2] [3] [16]. Also, the problem of multicast routing with NC has been addressed in [18] [19] [22] [23]. However, in these studies (especially the first one), the size of the field is associated with the solution, which is limited by the number of sinks that comprise the multicast network.

To make the field size independent of the number of sinks, this work proposes finding a multicast solution based on the order to distribute r-packets, which constitute the maximum flow of the multicast session between the outgoing links from the source node and guarantees the delivery of the r-packets to each of the sink nodes.

A code is defined to tag each packet within the network, which is within the range $[1, 2^r - 1]$. The original or so-called simple packets within this study are tagged with the vectors u_i in sequential order or with a defined mnemonic on an alphabet Σ. When packet coding is executed, it is also carrying out the linear combination of vectors u_i using coding coefficients with a value of 1 defined in $\mathbb{F}_2$ or with the representation of the mnemonics of the alphabet using the xor (+) operator.

From the matrix representation of the multicast session graph, the Reduced Source-Sink Hop matrix is constructed where each non-zero position indicates the number of times the outgoing link flow contributes at the incoming link to the constitution of a linear combination, which will enter the sink node at the end of this link.

A system of constraints is established that allows solutions with either simple packet ordering in outgoing links or the use of combined packets. These constraints are constructed from the Reduced Source-Sink Hop matrix and consider the linear independence conditions at the sink nodes to assume the resolution of the system. When the conditions of linear independence are not met, the method will generate systems without solutions.

A method is currently being developed to obtain all of the possible valid solutions for each case exposed. It is also expected that a proposal for multisource multicast (multicast multisession) systems, similar to this one, will be obtained. Furthermore, we are in the process of simulating a multicast session system that will allow us to observe its behavior beginning from the packets in the source node and include ordering in the outgoing links.

References

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, Vol. 46, No. 4, pp. 1204-1216, 2000. Article (CrossRef Link)

[2] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE ACM Trans. Netw. TON*, Vol. 11, No. 5, pp. 782-795, 2003. Article (CrossRef Link)

[3] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *Inf. Theory IEEE Trans.*, Vol. 49, No. 2, pp. 371-381, 2003. Article (CrossRef Link)

[4] P. A. Chou and Y. Wu, "Network coding for the internet and wireless networks," *IEEE Signal Process. Mag*., Vol. 24, No. 5, pp. 77-85, 2007. Article (CrossRef Link)

[5] B. Li and Y. Wu, "Network coding [scanning the issue]," *Proc. IEEE*, Vol. 99, No. 3, pp. 363-365, 2011. Article (CrossRef Link)

[6] A. Sprintson, "Network coding and its applications in communication networks," *Algorithms for Next Generation Networks*, Springer, pp. 343-372, 2010. Article (CrossRef Link)

[7] R. W. Yeung and N. Cai, Network coding theory. Now Publishers Inc, 2006. Article (CrossRef Link)

[8] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," *ACM SIGCOMM Comput. Commun. Rev.*, Vol. 36, No. 1, pp. 63-68, 2006. Article (CrossRef Link)

[9] R. W. Yeung, "Network coding: A historical perspective," *Proc. IEEE*, Vol. 99, No. 3, pp. 366-371, 2011. Article (CrossRef Link)

[10] P. Elias, A. Feinstein, and C. Shannon, "A note on the maximum flow through a network," *IRE Trans. Inf. Theory*, Vol. 2, No. 4, pp. 117-119, 1956. Article (CrossRef Link)

[11] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Classic papers in combinatorics*, pp. 243-248, 1987. Article (CrossRef Link)

[12] G. Dantzig and D. R. Fulkerson, "On the max flow min cut theorem of networks," *Linear Inequalities Relat. Syst*., Vol. 38, pp. 225-231, 2003. Article (CrossRef Link)

[13] R. Lidl and H. Niederreiter, Finite fields, vol. 20. Cambridge university press, 1997.

[14] R. Koetter and M. Médard, "Beyond routing: An algebraic approach to network coding," in *Proc. of INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Vol. 1, pp. 122-130, 2002. Article (CrossRef Link)

[15] S. Jaggi, P. A. Chou, and K. Jain, "Low complexity optimal algebraic multicast codes," in *Proc. of IEEE Int'l Symp. Information Theory*, Yokohama, Japan, 2003. Article (CrossRef Link)

[16] S. Jaggi et al., "Polynomial time algorithms for multicast network code construction," *Inf. Theory IEEE Trans*, Vol. 51, No. 6, pp. 1973-1982, 2005. Article (CrossRef Link)

[17] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," in *Proc. of the fifteenth annual ACM symposium on Parallel algorithms and architectures*, pp. 286-294, 2003. Article (CrossRef Link)

[18] Y. Zhu, B. Li, and J. Guo, "Multicast with network coding in application-layer overlay networks," *IEEE J. Sel. Areas Commun.*, Vol. 22, No. 1, pp. 107-120, 2004. Article (CrossRef Link)

[19] T. Shaoguo, H. Jiaqing, Y. Zongkai, R. S. Youail, and C. Wenqing, "Routing algorithm for network coding based multicast," in *Proc. of Convergence Information Technology International Conference*, pp. 2091-2095, 2007. Article (CrossRef Link)

[20] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, "Network flows: theory, algorithms, and applications," 1993.

[21] T. H. Cormen, Introduction to algorithms. MIT press, 2009.

[22] L. Wu and K. Curran, "A practical network coding and routing scheme based on maximum flow combination,” *Int. J. Netw. Manag.*, Vol. 22, No. 5, pp. 373-396, 2012. Article (CrossRef Link)

[23] D. Tang, X. Lu, y J. Li, "Multicast routing algorithm based on network coding," in *Proc. of International Conference on Brain Inspired Cognitive Systems*, pp. 348-357, 2013. Article (CrossRef Link)

[24] T. Ho and D. Lun, Network coding: an introduction. Cambridge University Press, 2008. Article (CrossRef Link)

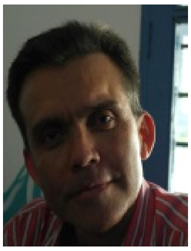
[25] K. Menger, "Zur allgemeinen Kurventheorie," *Fundam. Math.*, Vol. 10, No. 1, pp. 96-115, 1927. Article (CrossRef Link)

[26] S. I. Grossman, Elementary linear algebra. Brooks/Cole Publishing Company, 1994

[27] R. J. Wilson, An introduction to graph theory. Pearson Education India, 1970.

[28] "GitHub - Z3Prover/z3: The Z3 Theorem Prover," https://github.com/Z3Prover/z3

[29] S. Knight et al., "Virtual router redundancy protocol," 1998. https://tools.ietf.org/html/rfc2338



José Márquez Díaz received a MSc degree in Computer Science from ITESM-México in agreement with Universidad Autónoma de Bucaramanga and a BSc degree in Computer Science from Universidad del Norte, Barranquilla, Colombia. He has been a Full Professor since 1995 in the System and Computer Engineering Department at the Universidad del Norte. His main research is focused on the Quality of Service (QoS) in Computer Networks, AdHoc Networks, and Network Coding in Multicasting.

Ismael Gutiérrez García received a Ph.D. degree in Mathematics from Johannes Gutenberg University Mainz, Germany. He has been a Full Professor since 1995 in the Math Department at the Universidad del Norte. His current research interests are finite soluble groups, Discrete Mathematics, and their applications.

Sebastian Valle Herrera, is a Software Engineering student in the 10th semester from Universidad del Norte, Barranquilla, Colombia. His fields of interest include software security and network optimization**.**

Melanis Falco Pastrana, is a System Engineering student in the 10th semester from Universidad del Norte, Barranquilla, Colombia. She is especially interested in research in the networking technology area and security fields.