Network Coding based on quasigroups

Daniela Mechkaroska*, Aleksandra Popovska-Mitrovikj*, Verica Bakeva*, Vesna Dimitrova*
* Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Skopje, R.N. Macedonia
{daniela.mechkaroska, aleksandra.popovska.mitrovikj, verica.bakeva, vesna.dimitrova}@finki.ukim.mk

***Abstract*—Network coding is a promising technique that improves network throughput and provides high reliability. In this paper we propose an application of quasigroups in network coding and give several advantages of proposed algorithm over linear network coding.**

I. INTRODUCTION

Quasigroups and quasigroup transformations are very useful for construction of cryptographic primitives, error detecting and error correcting codes. The reasons for that are: the structure of quasigroups, their large number, the properties of quasigroup transformations ([1]) and others.

On the other side, the network coding is a technique which increases the network throughput and provides high reliability. Transmitted data is encoded in order to increase throughput, reduce delays and make the network more robust. In a node, algebraic algorithms are used to generate output messages by encoding its received messages. Then, the received transmissions are decoded at their destinations. On this way, fewer transmissions are required to transmit all data, but this requires more processing at intermediary and terminal nodes [2].

The most popular network coding is the linear network coding, where each node generates new packets that are linear combinations of earlier received packets, multiplying them by coefficients chosen from a finite field. Sink nodes receive these coded messages and collect them in a matrix. The original messages can be recovered by performing Gaussian elimination on the matrix [3][4][5].

In this paper, we will propose an application of quasigroup transformations in new algorithm for network coding in a butterfly network topology that provides simpler decoding process.

This paper is organized as follow. In Section II we give the basic definitions and properties of the quasigroups and quasigroups string transformation that will be used in this paper. In Section III, we explain a butterfly network like a form of network topology that can be used for connecting different nodes in a multiprocessor system and we present a new network coding algorithm based on quasigroups for increasing the throughput in butterfly communication network. At the end, some conclusions are made.

II. QUASIGROUP AND QUASIGROUP TRANSFORMATION

A quasigroup $(Q, *)$ is a groupoid, i.e., a set Q with a binary operation $*: Q^2 \to Q$, satisfying the law:

$$(\forall u, v \in Q)(\exists! x, y \in Q)(x * u = v \ \& \ u * y = v) \quad (1)$$

In fact, (1) says that a groupoid $(Q, *)$ is a quasigroup if and only if the equations $x * u = v$ and $u * y = v$ have unique solutions x and y for any given $u, v \in Q$.

The cardinality $|Q|$ of this set is called an order of the quasigroup. The main body of the multiplication table of a quasigroup is a Latin square over the set Q. Given a quasigroup $(Q, *)$, a new operation $\backslash$ (called a parastrophe) can be derived from the operation $*$ as follows:

$$x * y = z \Leftrightarrow y = x \backslash z \quad (2)$$

Then the algebra $(Q, *, \backslash)$ satisfies the identities:

$$x \backslash (x * y) = y \ \text{ and } \ x * (x \backslash y) = y \quad (3)$$

and $(Q, \backslash)$ is also a quasigroup.

Quasigroup string transformations are defined on a finite set Q (i.e., an alphabet Q) endowed with a quasigroup operation $*$, and they are mappings from Q^+ to Q^+, where $Q^+ = Q \cup Q^2 \cup Q^3 \cup \ldots$ is the set of all nonempty words on Q. In our algorithm, we use the following two quasigroup string transformations.

Let $l \in Q$ be a fixed element, called a leader. The transformations E_l and D_l are defined as follows:

$$E_{l,*}(a_1 a_2 \ldots a_n) = b_1 b_2 \ldots b_n \Leftrightarrow \begin{cases} b_1 = l * a_1 \\ b_i = b_{i-1} * a_i, \ 2 \le i \le n \end{cases}$$

$$D_{l,\backslash}(b_1 b_2 \ldots b_n) = a_1 a_2 \ldots a_n \Leftrightarrow \begin{cases} a_1 = l \backslash b_1 \\ a_i = b_{i-1} \backslash b_i, \ 2 \le i \le n \end{cases}$$

where $a_i, b_i \in Q$.

By using the identities (3) the following theorem is proved.

Theorem 1 [1] For each $M \in Q^+$ and each leader $l \in Q$ $D_{l,\backslash}(E_{l,*}(M)) = M = E_{l,*}(D_{l,\backslash}(M))$, i.e., $E_{l,*}$ and $D_{l,\backslash}$ are permutations on Q^+ mutually inverse. ■

For given quasigroup operations $*_1, *_2, \ldots, *_n$, on the set Q, the mappings $E_1, E_2, \ldots, E_n$ can be defined in the

Copyright 2020 by Information Society of Serbia - ISOS, Serbia | Creative Commons License: CC BY-NC-ND

This research was partially supported by Faculty of Computer Science and Engineering at the University "Ss Cyril and Methodius" in Skopje

same manner as previous, by choosing fixed elements l_1, l_2, …, $l_n \in Q$ (such that E_i is corresponding to $*_i$ and l_i). Let

$$E^{(n)} = E^{(n)}_{l_n,\ldots,l_1} = E_n \circ E_{n-1} \circ \ldots \circ E_1$$

where ° is the usual composition of mappings ($n \geq 1$).

Similarly, we can define composition $D^{(n)}$ of quasigroup transformations D_1, D_2, …, D_n, with leaders l_1, l_2, …, l_n respectively:

$$D^{(n)} = D^{(n)}_{l_1,\ldots,l_n} = D_1 \circ D_2 \circ \ldots \circ D_n$$

The transformations $E^{(n)}$ and $D^{(n)}$ defined on this way are mutually inverse. This allows us to use $E^{(n)}$ − transformation for coding and encrypting of messages and $D^{(n)}$ − transformation for decoding and decrypting. Namely, applications of a quasigroup transformation on string provides many useful cryptographic properties. First of all, if M is the original message and $C = E^{(n)}(M)$ is the message obtained after application of $E^{(n)}$ on M, then it is impossible to find the original message M from C without knowing all quasigroups in $E^{(n)}$. The brute-force attack for finding these quasigroups is impossible due the large number of quasigroups. Further on, in [1] the authors proved that statistical kind of attack is also impossible. It follows from the proof of the following theorem.

Theorem 2 [1] Let $M \in Q+$ be an arbitrary string of length l and $C = E^{(n)}(M)$. Then m-tuples in C are uniformly distributed for any $m \leq n$, for enough large l. ■

Further on, we will propose an application of these quasigroup transformations in network coding.

III. New Network Coding Algorithm Based on Quasigroup Transformations

In this paper we propose a new network coding algorithm based on quasigroup transformations for increasing the throughput in butterfly communication network. This algorithm is simpler and faster than linear network coding.

A butterfly network is a form of network topology that can be used for connecting different nodes in a multiprocessor system [6]. This network is often used to illustrate how network coding can outperform routing. Two source information M^1 and M^2 (Fig. 1) must be transmitted to two destination nodes R_1 and R_2. Each node can carry only single information. The central link (from node B to node D) would be able to carry only M^1 or M^2, but not both. Suppose we send M^1 through the center, then R_1 would receive M^1 twice and not receive M^2 at all. Sending M^2 poses a similar problem for R_2. So, the routing is insufficient because no routing scheme can transmit both M^1 and M^2 simultaneously to both destinations.

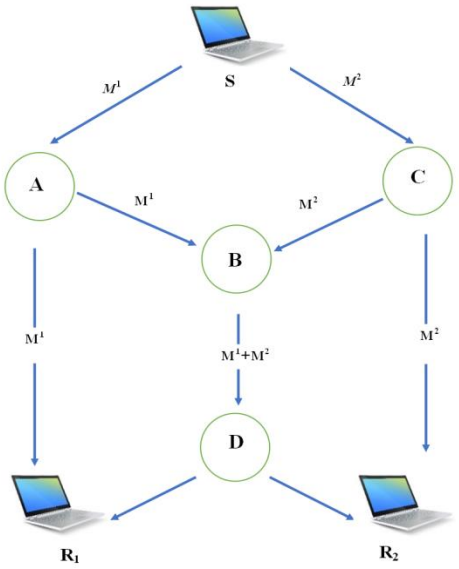

Figure 1. Butterfly network

Our idea for network coding based on quasigroups is presented on a simple network shown on Fig. 2. In this algorithm we use:

- Quasigroup $(Q, *)$ of order 2^n together with its parastrophe \.
- $E_{l,*}$ - transformation with leader l and quasigroup operation * (for coding and encrypting of messages) and $D_{l,\backslash}$ - transformation with the same leader l and corresponding quasigroup operation \ (for decoding and decrypting).

Network Coding Algorithm

At the source S, a finite amount of information is generated and multicast to other nodes on the network. Let M^1 and M^2 be the original messages of m symbols from Q.

Step 1. The node A receives the message M^1 and it sends the encoded message $E_{l1}(M^1)$ to the intermediate node B and to the receiver R_1. In the same way, the node C receives the messages M^2 and it sends the encoded message $E_{l2}(M^2)$ to the intermediate node B and to the receiver R_2.

Step 2. The intermediate node B makes an addition modulo 2^n of two received messages $E_{l1}(M^1)$ and $E_{l2}(M^2)$ and sends the obtained message b to node D.

Step 3. The node D sends encoded message $E_{l3}(b)$ to receivers R_1 and R_2. So, the node R_1 receives $E_{l1}(M^1)$ and $E_{l3}(b)$, and node R_2 receives $E_{l2}(M^2)$ and $E_{l3}(b)$. ■

Destinations nodes, R_1 and R_2, decoded original information M^1 and M^2 applying the following algorithm.

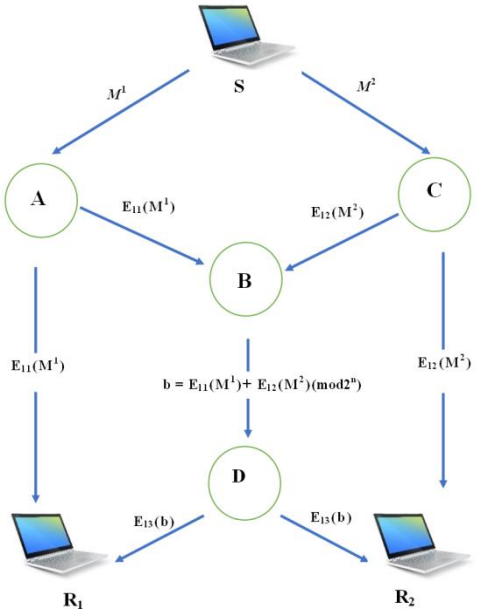

Figure 2. Network coding based on quasigroup

Decoding Algorithm

In node R_1:

The received messages are $E_{l1}(M^1)$ and $E_{l3}(b)$.

Step 1. The message b is obtained by applying D_{l3} on $E_{l3}(b)$, i.e., $b = D_{l3}(E_{l3}(b))$.

Step 2. The message $E_{l2}(M^2)$ is obtained with subtraction modulo 2^n of b and $E_{l1}(M^1)$.

Step 3. The original messages M^1 and M^2 are obtained by applying the inverse transformations D_{l1} and D_{l2} on $E_{l1}(M^1)$ and $E_{l2}(M^2)$, correspondingly, i.e., $M^1 = D_{l1}(E_{l1}(M^1))$ and $M^2 = D_{l2}(E_{l2}(M^2))$.

In note R_2:

The received messages are $E_{l2}(M^2)$ and $E_{l3}(b)$.

Step 1. The message b is obtained by applying D_{l3} on $E_{l3}(b)$, i.e., $b = D_{l3}(E_{l3}(b))$

Step 2. The message $E_{l1}(M^1)$ is obtained with subtraction modulo 2^n of b and $E_{l2}(M^2)$.

Step 3. The original messages M^1 and M^2 are obtained by applying the inverse transformations D_{l1} and D_{l2} on $E_{l1}(M^1)$ and $E_{l2}(M^2)$, correspondingly, i.e., $M^1 = D_{l1}(E_{l1}(M^1))$ and $M^2 = D_{l2}(E_{l2}(M^2))$. ■

Notice that the corresponding steps in note R_1 and R_2 can be performed in parallel.

Next, we will give an example for process of coding and decoding with this algorithm using quasigroup of order 2^2.

Example. Let $Q = \{0, 1, 2, 3\}$. We consider the following quasigroup operation * and the corresponding operation \ on Q.

*	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

\	0	1	2	3
0	0	1	2	3
1	3	0	1	2
2	2	3	0	1
3	1	2	3	0

- Coding

If $M^1 = 1233$, $M^2 = 2323$, $l_1 = 1$, $l_2 = 2$ and $l_3 = 3$, then node A sends $E_1(1233) = 2032$ to B and R_1, C sends $E_2(2323) = 0310$ to B and R_2.

Node B receives $E_1(1233) = 2032$, $E_2(2323) = 0310$ and sends $b = (2032+0310) \bmod 4 = 2302$ to D. Then, D sends $E_3(2302) = 1002$ to R_1 and R_2.

- Decoding

R_1 receives:

$E_1(1233) = 2032$
$E_3(2302) = 1002$

Source messages are obtained with the following operations:

$b = D_3(1002) = 2302$

$E_2(2323) = (2302 - E_1(1233)) \bmod 4$
$= (2302 - 2032) \bmod 4 = 0310$

$M^1 = D_1(2032) = 1233$

$M^2 = D_2(0310) = 2323$

R_2 receives:

$E_2(2323) = 0310$

$E_3(2302) = 1002$

Source messages are obtained with the following operations:

$b = D_3(1002) = 2302$

$E_1(1233) = (2302 - E_2(2323)) \bmod 4$
$= (2302 - 0310) \bmod 4 = 2032$

$M^1 = D_1(2032) = 1233$

$M^2 = D_2(0310) = 2323$

IV. CONCLUSION

In this paper we propose new algorithm for network coding. This algorithm has several advantages over linear network coding. The first one is that decoding uses much simpler operations than in linear network coding. Namely, instead of solving a system of linear equations using method of Gaussian elimination, we apply only a *D*-transformation. Therefore, our algorithm provides simpler and faster decoding process.

Also, due to the already proved cryptographic properties of used quasigroups transformations this algorithm provides an information security of transmitted data.

For further research we can consider how this algorithm can be applied for transmission in more complicated network topologies.

REFERENCES

[1] S. Markovski, D. Gligoroski, and V. Bakeva, "Quasigroup string processing: Part 1", Contributions, Sec. Math. Tech. Sci., MANU, vol. 20 1-2, 1999, pp.13–28.

[2] F. A. Monteiro, A. Burr, I. Chatzigeorgiou, I. *et al.*, "Special issue on network coding", *EURASIP J. Adv. Signal Process.* 2017**,** 29 (2017). https://doi.org/10.1186/s13634-017-0463-2.

[3] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding", Allerton Conference on Communication, Control, and Computing, October 2003.

[4] S.Y. R. Li, R. W. Yeung, and N. Cai, "Linear Network Coding", IEEE Transactions on Information Theory, Vol. 49, No. 2, pp. 371–381, February 2003.

[5] R. Koetter, and M. Médard, "An Algebraic Approach to Network Coding", IEEE/ACM Transactions on Networking, Vol. 11, No. 5, pp. 782–795, October 2003.

[6] Ch.Fragouli, E.Soljanin, "Network Coding Fundamentals", Foundations and Trends in Networking, Vol. 2, No. 1, 2007, 1–133, 2007.