Solving the Shortest Path Problem Using Integer Residual Arithmetic

Victor Krasnobayev
Electronics and Control Systems Department
V. N. Karazin Kharkiv National University
Kharkiv, Ukraine
v.a.krasnobaev@gmail.com

Alexandr Kuznetsov
Department of Information Systems and Technologies Security V. N. Karazin Kharkiv National University,
Kharkiv, Ukraine
kuznetsov@karazin.ua

Alina Yanko
Department of Computer and Information Technologies and Systems
National University «Yuri Kondratyuk Poltava Polytechnic», Poltava, Ukraine
al9_yanko@ukr.net

Tetiana Kuznetsova
Department of Information Systems and Technologies Security V. N. Karazin Kharkiv National University,
Kharkiv, Ukraine
kuznetsova.tatiana17@gmail.com

***Abstract*—The report considers solution to the problem of routing, the essence of which is to determine the shortest path length between any pair of computer network subscribers represented as an undirected graph, as one of the possible methods to increase the speed and performance of computer systems (CS). To carry out calculations and comparative analysis of the speed and productivity of CS in a positional binary number system (PNS) and in a non-positional number system in residual classes (residual number system – RNS), we consider one practical problem. Task is the routing problem, the essence of which is to determine the shortest path length, that is, to find the optimal data transmission route in the computer network.**

***Keywords*—binary number syste, computer network, positional number system, residual number system, routing algorithm, shortest path length, undirected graph**

I. Introduction

For a modern computer system (CS), the basic characteristics of a computer can be distinguished: performance; reliability and resiliency; scalability; software compatibility and portability. The main task of the research is calculation and comparative analysis of the performance of CS operating in a RNS and in a binary PNS will be carried out using the example of solving the problem of determining the optimal (shortest) path length of a computer network (CN). System performance will be considered as the amount of computational work performed per unit time. The most important parameter CS performance is the speed of execution of operations of a certain class, in our case the speed of determining the optimal network path [1]–[3]. The essence of this problem is to determine the optimal (shortest) path length between any pair of nodes of the CN, presented in the form of an undirected graph [4], [5]. There are currently many effective routing algorithms. Routing algorithms can be differentiated based on several key characteristics. First, the operation of the resulting routing protocol is influenced by specific tasks that the algorithm developer solves. Second, there are different types of routing algorithms, and each affects network and routing resources differently. Finally, routing algorithms use the various indicators that influence the calculation of optimal routes.

In this task will look at a routing algorithm based on tables (called vectors or matrices), supported by all routers and containing information about the shortest paths to each of the possible network nodes. The problem is solved by the matrix method [3], [6]–[8], which requires knowledge of the structure of the telecommunication network (TN) as a whole and therefore is most often used in centralized management. It defines a distance (solution matrix, shortest path matrix, dispersion matrix) matrix.

This work was supported in part by the National Research Foundation of Ukraine under Grant 2020.01/0351.

II. The Problem of Determining the Shortest Path Length

With the matrix method, operations are performed on the matrix L of lengths of network branches, and the optimized parameter is the path length between any pair of nodes in the graph, where the original matrix of path lengths is represented as [1], [2], [9], [10]: $L^{(1)} = \left|l_{ij}\right|^{(1)}$, where the numerical value of an arbitrary element $l_{ij}^{(1)}$ of the original matrix L determines, in arbitrary units, the distance between nodes i and j.

If the nodes i and j are not adjacent (there is no edge), then $l_{ij} = \infty$. The distance inside the node is considered zero $\left(l_{ij} = 0\right)$. The length of the shortest paths between all pairs of network nodes can be determined by using the so-called operation "raising the original matrix $\boldsymbol{L}^{(1)}$ to the R-th power" $(R \le (N-1)$, i.e. the number of iterations R cannot exceed the value $N-1$ (N is the number of nodes in the graph).

The usual raising of a matrix to a power is a sequential multiplication of it by itself $(R-1)-n$ times (multiplication "on the left", i.e. $L^{(R)} = L^{(1)} \cdot L^{(R-1)}$).

978-x-xxxx-xxxx-x/20/$31.00 ©2020 IEEE

If $l_{im}^{(1)}$ is the element of the i-th row and m-th column of the matrix $L^{(1)}$, and $l_{mj}^{(R-1)}$ is the element of the m-th row and j-th column of the matrix $L^{(R-1)}$, then, according to the matrix multiplication rule, the element $l_{ij}^{(R^*)}$ of the matrix $L^{(R)}$ is:

$$l_{ij}^{(R^*)} = l_{i1}^{(1)} \cdot l_{1j}^{(R-1)} + l_{i2}^{(1)} \cdot l_{2j}^{(R-1)} + \ldots + \\ + l_{im}^{(1)} \cdot l_{mj}^{(R-1)} + \ldots + l_{iN}^{(1)} \cdot l_{Nj}^{(R-1)}, \qquad (1)$$

In the case of solving the routing problem in expression (1) to determine the value $l_{ij}^{(R^*)}$ we replace the operation of "multiplication" by addition, and the operation of "addition" by choosing the minimum from the obtained values of the sums [4], [5], [11], [12]:

$$l_{ij}^{(R)} = \min\left\{\sum_{i,j}^{N}\left(l_{im}^{(1)} + l_{mj}^{(R-1)}\right)\right\},\ m = \overline{1,\ N}, \qquad (2)$$

We show that the value $l_{in}^{(R)}$ is equal to the "weight" x_{in} of node i with respect to the selected node n, i.e. minimum distance to it. Indeed,

$$\begin{cases} x_{in}^{(1)} = l_{in}^{(1)}, \\ x_{in}^{(2)} = \min\limits_{m}\left(l_{im}^{(1)} + x^{(1)}{}_{mn}\right) = \min\limits_{m}\left(l_{im}^{(1)} + l^{(1)}{}_{mn}\right) = l_{in}^{(2)}, \\ \vdots \\ x_{in}^{(R)} = \min\limits_{m}\left(l_{im}^{1} + x_{mn}^{R-1}\right) = \min\limits_{m}\left(l_{im}^{(1)} + l^{(R-1)}{}_{mn}\right) = l_{in}^{(R)}. \end{cases} \qquad (3)$$

Therefore, the value $l_{ij}^{(R)}$, at $R \le N-1$ (R is the number of traversable branches of the graph) is the length of the shortest path between node i and node j of the TN.

For a particular network, the matrix $L^{(R)}$ for which $L^{(R)} = L^{(R-1)}$ is calculated. In this case, we obtain a matrix of solutions [2], [5], [13]. After that, the calculations stop.

As you can see, the basic operation (2) for this integer algorithm consists of two main operations of addition and comparison. The total number of basic type operations:

$$c_{ij} = \sum\nolimits_{i,j}^{N}\left(a_{ik} \cdot b_{kj}\right) = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + a_{i3} \cdot b_{3j} + \\ + \ldots + a_{iN} \cdot b_{Nj}\ ,\ \left(i, j = \overline{1, N}\right), \text{ is } N^2 \ .$$

Note that the presented matrix optimization method can be used not only to find the path of the minimum length, but also the most reliable path - the path with the minimum value of the probability of message loss [14]–[17]. For this, it is necessary to find such a path in which the product of the probabilities ω_{ij} of the serviceable state of its individual sections l_{ij} has a maximum value:

$$P_K = \max \prod_{\rho=1}^{n} \omega_{ij}\ , \qquad (4)$$

where n is the number of transit sections of the K-th along the path $K = (1, 2, 3\ldots)$ of the CS.

III. The Method of Fast Implementation of the Routing Algorithm

Based on the use of formula (2), we consider the method of fast (accelerated) implementation of the routing algorithm. In this case, all operations of addition of the form $\left\{l_{im}^{(1)} + l_{mj}^{(R-1)}\right\}$ (see Fig. 1) are combined in time.

We use a residual class system to implement integer arithmetic [18]–[21]. This mathematical apparatus can significantly reduce the complexity of calculations in many applications of computer science [22]–[26]. Among other things, the use of a residual class system allows you to implement error correction [27]–[29], improve reliability and fault tolerance [30]–[33], and much more.

The time $T_{rout.}^{(PNS)}$ to solve the routing algorithm in the PNS (determining the length of the shortest path between an arbitrary pair of graph nodes) in accordance with the expression (2), (3) is determined by the formula:

$$T_{rout.}^{(PNS)} = \tau_{add.}^{(PNS)} + N_{comp.} \cdot \tau_{comp.}^{(PNS)} = (2 \cdot \rho - 1) \cdot \tau + \left(\right]\log_2 N\left[\right) \cdot 5\tau/2$$

or

$$T_{rout.}^{(PNS)} = \frac{\left[2 \cdot (2\rho - 1) + 5\left(\right]\log_2 N\left[\right)\right] \cdot \tau}{2} \qquad (5)$$

Using expressions (2), (3), we determine the time $T_{rout.}^{(RNS)}$ for the implementation of the routing algorithm in the RNS, which will be presented in the form of the expression:

$$T_{rout.}^{(RNS)} = \tau_{comp.}^{(RNS)} + N_{comp.} \cdot \tau_{comp.}^{(RNS)} = \tau_T + N_{comp.} \cdot \tau_{comp.}^{(RNS)} = \\ = \tau_{AND} + N_{comp.} \cdot \tau_{comp.}^{(RNS)} = \tau/2 + \left(\right]\log_2 N\left[\right) \cdot 5 \cdot \tau$$

or

$$T_{rout.}^{(RNS)} = \frac{\left\{1 + \left(\right]\log_2 N\left[\right) \cdot 10\right\} \cdot \tau}{2}, \qquad (6)$$

The effectiveness $K_{eff.}^{(l)}$ of using RNS to improve the performance of the routing algorithm implementation can be determined by the expression increase data processing speed increases.

$$K_{eff.}^{(l)} = T_{rout.}^{(PNS)} / T_{rout.}^{(RNS)} = \frac{2 \cdot (2 \cdot \rho - 1) + 5 \cdot \left(\right]\log_2 N\left[\right)}{1 + 10 \cdot \left(\right]\log_2 N\left[\right)}, \qquad (7)$$

The results of the calculation and comparative analysis of the effectiveness of the use of RNS are presented in table I.

An analysis of the results of calculations and studies of the performance of CS operating in RNS and in binary PNS showed the following. From the point of view of increasing the performance of the CS, it is preferable to use RNS as a number system.

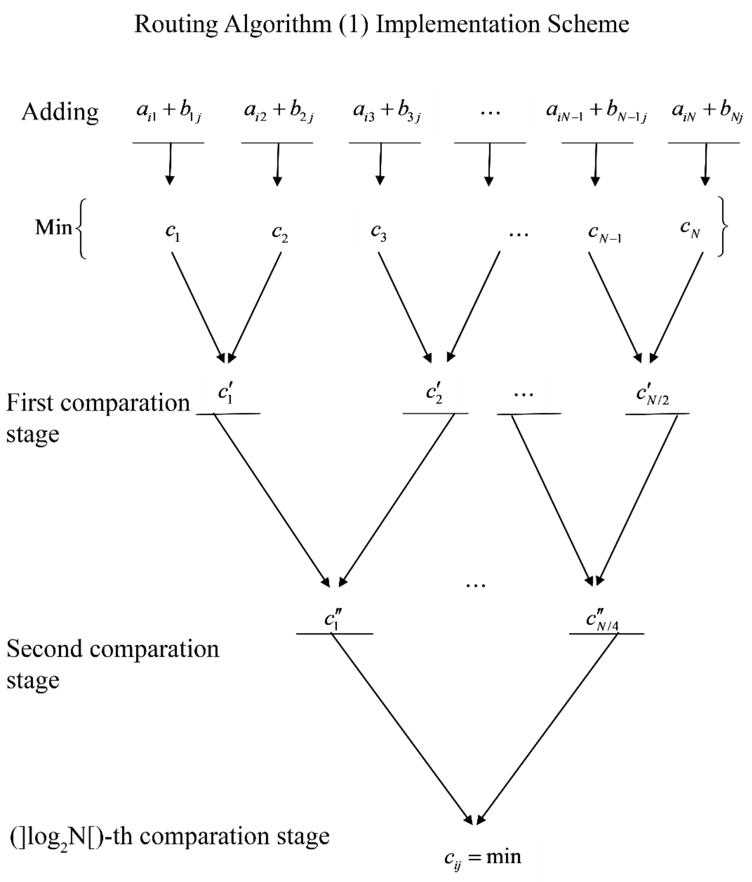

Fig. 1. Routing algorithm implementation scheme

TABLE I. ESTIMATED DATA AND COMPARATIVE ANALYSIS OF THE EFFECTIVENESS OF USING RNS

N	$l=1$		
	$T_{rout.}^{(PNS)}$	$T_{rout.}^{(RNS)}$	$K_{eff.}^{(l)}$
4	20	10,5	1,9
8	22,5	15,5	1,4
10	25	20,5	1,2
N	$l=2$		
	$T_{rout.}^{(PNS)}$	$T_{rout.}^{(RNS)}$	$K_{eff.}^{(l)}$
4	36	4	36
8	38,5	8	38,5
10	41	10	41
N	$l=3$		
	$T_{rout.}^{(PNS)}$	$T_{rout.}^{(RNS)}$	$K_{eff.}^{(l)}$
4	52	4	52
8	54,5	8	54,5
10	57	10	57
N	$l=4$		
	$T_{rout.}^{(PNS)}$	$T_{rout.}^{(RNS)}$	$T_{rout.}^{(PNS)}$
4	68	4	68
8	70,5	8	70,5
10	73	10	73
N	$l=8$		
	$T_{rout.}^{(PNS)}$	$T_{rout.}^{(RNS)}$	$T_{rout.}^{(PNS)}$
4	132	4	132
8	134	8	134
10	137	10	137

CONCLUSION

The report clarifies some aspects of the theory of graph. For calculations and comparative analysis of the speed and productivity of computer systems (CS) in a positional binary number system (PNS) and in a non-positional number system in residual classes, we considered practical problem is optimal routing problem. A method of fast (accelerated) implementation of the routing algorithm has been developed. The calculated data and a comparative analysis of the efficiency of increasing the performance of the routing algorithm implementation using RNS and a positional binary number system are carried out. The effectiveness of using RNS to increase the productivity of routing algorithm implementation has been proved by real calculations.

REFERENCES

[1] V. E. Muhin and W. B. Hu, Eds., "A Shortest Path Problem on the Network with Stochastic Arc Length," in *International Conference on Information Technology and Computer Science, 3rd (ITN 2011)*, ASME Press, 2011, pp. 378–381.

[2] G. Kumar, R. K. Bajaj, and N. Gandotra, "Algorithm for Shortest Path Problem in a Network with Interval-valued Intuitionistic Trapezoidal Fuzzy Number," *Procedia Comput. Sci.*, vol. 70, pp. 123–129, Jan. 2015, doi: 10.1016/j.procs.2015.10.056.

[3] D. M. L. Dias Rasteiro, "9 - Shortest path problem and computer algorithms," in *Calculus for Engineering Students*, J. Martín-Vaquero, M. Carr, A. Queiruga-Dios, and D. Richtáriková, Eds. Academic Press, 2020, pp. 179–195.

[4] A. Nakayama and T. Anazawa, "Dijkstra-Based Algorithms for the Shortest Path Problem with Edges of Negative Length," *J. Oper.*

Res. Soc. Jpn., vol. 56, no. 2, pp. 137–154, 2013, doi: 10.15807/jorsj.56.137.

[5] S. A. Bakar and M. Ibrahim, “Optimal solution for travelling salesman problem using heuristic shortest path algorithm with imprecise arc length,” *AIP Conf. Proc.*, vol. 1870, no. 1, p. 040061, Aug. 2017, doi: 10.1063/1.4995893.

[6] C. Rao and L. Yu, “A particular method for the shortest path finding of mobile robots,” in *2010 The 2nd International Conference on Industrial Mechatronics and Automation*, May 2010, vol. 2, pp. 221–224, doi: 10.1109/ICINDMA.2010.5538330.

[7] S. Goto, T. Ohtsuki, and T. Yoshimura, “Sparse matrix techniques for the shortest path problem,” *IEEE Trans. Circuits Syst.*, vol. 23, no. 12, pp. 752–758, Dec. 1976, doi: 10.1109/TN.1976.1084155.

[8] A. Brodnik and M. Grgurovič, “Practical Algorithms for the All-Pairs Shortest Path Problem,” in *Shortest Path Solvers. From Software to Wetware*, A. Adamatzky, Ed. Cham: Springer International Publishing, 2018, pp. 163–180.

[9] A. K. Nemani and R. K. Ahuja, “Shortest Path Problem Algorithms,” in *Wiley Encyclopedia of Operations Research and Management Science*, American Cancer Society, 2011.

[10] Ji-Xian Xiao and Fang-Ling Lu, “An improvement of the shortest path algorithm based on Dijkstra algorithm,” in *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, Feb. 2010, vol. 2, pp. 383–385, doi: 10.1109/ICCAE.2010.5451564.

[11] M. Helaoui, “Extended Shortest Path Problem - Generalized Dijkstra-Moore and Bellman-Ford Algorithms,” Aug. 2020, pp. 306–313, Accessed: Aug. 16, 2020. [Online]. Available: https://www.scitepress.org/Link.aspx?doi=10.5220/0006145303060313.

[12] Y. Gupta, M. Yamuna, and C. Saha, “Finding shortest path between two locations using matrix representation,” in *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)*, May 2014, pp. 1–5, doi: 10.1109/ICRAIE.2014.6909250.

[13] K. Ambrus-Somogyi, A. Bakó, P. Földesi, and J. Gáti, “Fast algorithm to solve the most economical path problem in sparse matrices,” in *2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES)*, Jun. 2012, pp. 335–339, doi: 10.1109/INES.2012.6249854.

[14] “Advanced Protocols for Wireless Ad-hoc Networks.” https://ieeexplore.ieee.org/courses/details/EDP036 (accessed Aug. 16, 2020).

[15] E. Carlos Gomes Wille and M. Mincov Tenorio, “Considering Packet Loss Probability in Fault-Tolerant OSPF Routing,” *IEEE Lat. Am. Trans.*, vol. 12, no. 2, pp. 248–255, Mar. 2014, doi: 10.1109/TLA.2014.6749545.

[16] A. Mellouk, S. Hoceini, F. Baguenine, and M. Cheurfa, “Distributed Quality-of-Service routing of best constrained shortest paths.,” in *2008 IEEE Symposium on Computers and Communications*, Jul. 2008, pp. 915–919, doi: 10.1109/ISCC.2008.4625670.

[17] J. Wu, N. Liu, S.-K. Lam, and G. Jiang, “Shortest Partial Path First Algorithm for Reconfigurable Processor Array with Faults,” in *2016 IEEE Trustcom/BigDataSE/ISPA*, Aug. 2016, pp. 1198–1203, doi: 10.1109/TrustCom.2016.0194.

[18] J. O. Tuazon, “Residue number system in computer arithmetic,” Doctor of Philosophy, Iowa State University, Digital Repository, Ames, 1969.

[19] I. Koren, “THE RESIDUE NUMBER SYSTEM,” *Computer Arithmetic Algorithms*, Oct. 08, 2018. https://www.taylorfrancis.com/ (accessed Aug. 16, 2020).

[20] P. V. Ananda Mohan, “Specialized Residue Number Systems,” in *Residue Number Systems: Theory and Applications*, P. V. A. Mohan, Ed. Cham: Springer International Publishing, 2016, pp. 177–193.

[21] P. V. A. Mohan, *Residue Number Systems*. Cham: Springer International Publishing, 2016.

[22] Y. Zhang, “An FPGA implementation of redundant residue number system for low-cost fast speed fault-tolerant computations,” Thesis, 2018.

[23] V. Krasnobayev, A. Kuznetsov, A. Yanko, B. Akhmetov, and T. Kuznetsova, “Processing of the Residuals of Numbers in Real and Complex Numerical Domains,” in *Data-Centric Business and Applications*, vol. 48, T. Radivilova, D. Ageyev, and N. Kryvinska, Eds. Cham: Springer International Publishing, 2021, pp. 529–555.

[24] P. V. An, a Mohan, CDAC, and Bangalore, “Implementation of Residue Number System Based Digital Filters – A Quarterly Publication of ACCS.” https://journal.accsindia.org/implementation-of-residue-number-system-based-digital-filters/ (accessed Aug. 16, 2020).

[25] V. Krasnobayev, A. Kuznetsov, V. Popenko, A. Kononchenko, and T. Kuznetsova, “Determination of Positional Characteristics of Numbers in the Residual Class System,” in *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, May 2020, pp. 151–156, doi: 10.1109/DESSERT50317.2020.9125030.

[26] S. Timarchi and K. Navi, “Efficient Class of Redundant Residue Number System,” in *2007 IEEE International Symposium on Intelligent Signal Processing*, Oct. 2007, pp. 1–6, doi: 10.1109/WISP.2007.4447506.

[27] D. I. Popov and A. V. Gapochkin, “Development of Algorithm for Control and Correction of Errors of Digital Signals, Represented in System of Residual Classes,” in *2018 International Russian Automation Conference (RusAutoCon)*, Sep. 2018, pp. 1–3, doi: 10.1109/RUSAUTOCON.2018.8501826.

[28] V. Krasnobayev, A. Kuznetsov, A. Yanko, and K. Kuznetsova, “Correction Codes in the System of Residual Classes,” in *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S T)*, Oct. 2019, pp. 488–492, doi: 10.1109/PICST47496.2019.9061253.

[29] V. Krasnobayev, A. Kuznetsov, A. Yanko, and K. Kuznetsova, “The data errors control in the modular number system based on the nullification procedure,” in *Proceedings of The Third International Workshop on Computer Modeling and Intelligent Systems (CMIS-2020), Zaporizhzhia, Ukraine, April 27-May 1, 2020*, 2020, vol. 2608, pp. 580–593, Accessed: Jul. 02, 2020. [Online]. Available: http://ceur-ws.org/Vol-2608/paper45.pdf.

[30] D. Radhakrishnan and T. Pyon, “Fault tolerance in RNS: an efficient approach,” in *Proceedings., 1990 IEEE International Conference on Computer Design: VLSI in Computers and Processors*, Sep. 1990, pp. 41–44, doi: 10.1109/ICCD.1990.130156.

[31] R. Forsati, K. Faez, F. Moradi, and A. Rahbar, “A Fault Tolerant Method for Residue Arithmetic Circuits,” in *2009 International Conference on Information Management and Engineering*, Kuala Lumpur, Malaysia, 2009, pp. 59–63, doi: 10.1109/ICIME.2009.111.

[32] N. I. Cherviakov, M. G. Babenko, and M. N. Shabalina, “Development of a secure system for distributed data storage and processing in the clouds based on the concept of active security in RNS,” in *2017 XX IEEE International Conference on Soft Computing and Measurements (SCM)*, May 2017, pp. 558–560, doi: 10.1109/SCM.2017.7970649.

[33] V. Krasnobayev, A. Kuznetsov, A. Yanko, and T. Kuznetsova, “The analysis of the methods of data diagnostic in a residue number system,” in *Proceedings of The Third International Workshop on Computer Modeling and Intelligent Systems (CMIS-2020), Zaporizhzhia, Ukraine, April 27-May 1, 2020*, 2020, vol. 2608, pp. 594–609, Accessed: Jul. 02, 2020. [Online]. Available: http://ceur-ws.org/Vol-2608/paper46.pdf.