Al-Salam Journal for Engineering and Technology
Journal Homepage: http://journal.alsalam.edu.iq/index.php/ajest
e-ISSN: 2790-4822p-ISSN: 2958-0862

A New Fractal Topologies Based on Hypercube Interconnection Network

Muslim Mohsin Khudhair[1,2*], Furkan Rabee[1], Adil Al-Rammahi[3]

[1]Department of Computer Science, Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq.
[2]Department of Computer Information System, College of Computer Science and Information Technology, University of Basrah, Basrah, Iraq.
[3]Department of Mathematical Science, Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq.

*Corresponding Author: Muslim Mohsin Khudhair

DOI: https://doi.org/10.55145/ajest.2023.02.02.016
Received January 2023Accepted March 2023; Available online April 2023

ABSTRACT: A parallel processing system's most crucial part is a network interconnection that links its processors. The hypercube topology has exciting features, making it an excellent option for parallel processing applications. This paper presents two innovative configurations of interconnection networks based on fractal Sierpinski and a hypercube. These are the Sierpinski Triangle Topology (STT) and Sierpinski Carpet Topology (SCT). Compared to a hypercube, the Sierpinski Triangle topology (STT) noticed a significant decrease in the number of nodes and links as large networks grew. Hence, it considers a great way to reduce costs because it uses fewer nodes and links. The average distance is also shorter, which is better. Despite it has a smaller bisection width and a more degree than a hypercube by one. The Sierpinski Carpet Topology (SCT) has the advantage of having a higher bisection width than a hypercube. That is preferable because it places a lower restriction on the difficulty of parallel algorithms. In contrast, the drawback of this topology is that it has a diameter and average distance more than a hypercube.

Keywords: Interconnection network, Hypercube, Fractal, Sierpinski Triangle, Sierpinski Carpet

1. INTRODUCTION

The multicore processor technology is the foundation of any high-performance computer system today. Multicore processors are used in everything from basic embedded systems to advanced server farms. Such multicore systems' performance depends on the interconnection network connecting these cores [1]. The need to develop parallel processing in computers has become essential, leading to the advent of newer interconnection networks to enable parallel processing. Interconnection networks, abbreviated as (Ins), may be classified as either dynamic or static [2].

Connections in a static network are permanent ties, but connections in a dynamic network may be built up on the fly according to the system’s requirements. Point-to-point connections directly link a processor and other processors in static networks. It is also possible to classify it further according to the connectivity pattern as either having one dimension (1D), two dimensions (2D), or a hypercube (HC). In contrast, according to the interconnection methods, dynamic networks may be divided into bus-based and switch-based categories. Two subcategories may be applied to bus-based networks: single buses and multiple buses. According to the nature of the interconnection network, switch-based dynamic networks may be classified as single-stage (SS), multistage (MS), or crossbar networks [1, 3]. The primary categories of interconnection networks display in Fig. 1.

Interconnection networks may also be divided into electrical and optoelectronic communication channels connecting processors. Examples are hypercube, mesh, ring, tree, and other electronic connectivity networks. Optical Chained-Cubic Tree (OCCT) and Optical Transpose Interconnection System (OTIS) are examples of optoelectronic interconnection networks. For OTIS, there are several architectures, including OTIS Hyper Hexa-Cell (OHHC), OTIS-Hypercube, OTIS Mesh, and many more [4].

**Corresponding author: muslim.khudhair@uobasrah.edu.iq*

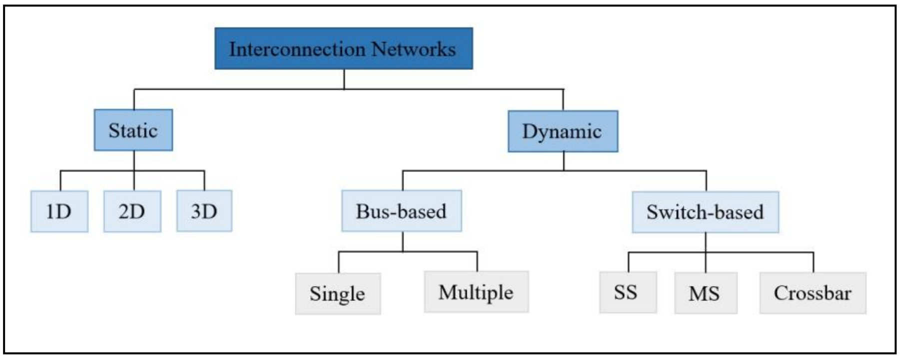

FIGURE 1. - Topology-based classification of interconnection networks [1]

The interconnection networks compare in terms of several topological properties, the most important of which are [5, 6]:

1. The diameter of a network is the length of the shortest path between a source and a destination.
2. The degree of a node reveals how many nodes directly connect to it.
3. The area cost of a network determines by the diameter and the degree.
4. The average distance is the sum of the distances of all nodes from a particular node (the source) divided by the number of nodes. The average distance is critical in determining the overall delay in a computer network.
5. The bisection width is the smallest number of channels that must be cut to split the network into two identical groups. The bisection width measures the network's communication bottleneck when random traffic is spread out evenly.
6. A network is node symmetric if each switch has the same channel properties. Communication can be a bottleneck at asymmetric switches.

The paper is organized as follows. This introduction is followed by section 2, which discusses the related work connected to this topic. Next, section 3 describes the detail of the hypercube, which regards a basis for the proposed topologies. Section 4 explains the introductory presentation of fractals and their shapes that are of interest within the proposed research. The proposed approach of the Sierpinski Triangle Topology (STT) and Sierpinski Carpet Topology (SCT) is presented in section 5. Section 6 deals with performance evaluation, and section 7 includes the conclusion.

2. RELATED WORK

Parallel interconnection networks, which serve as the backbone of parallel systems, have become a foundation for study. Planning a new interconnection network and determining its cost-efficiency will take priority at this stage. Additionally, quicker connectivity is required to achieve faster computation. Several strategies use to accomplish the desired improvement, and each methodology has produced a new topology. The performance metrics are examined to demonstrate their superiority [6, 7].

The hypercube has emerged as a practical option among researchers due to its effective attributes, including a regular and symmetric architecture, a small diameter, a low node degree, and a low link complexity. These characteristics have contributed to the hypercube's rise to the top of the popularity rankings [8]. Over the years, many scholars have spent time and energy researching connectivity networks. Interconnection networks have severe constraints due to their topological structures, and the drawbacks include a high cost and a large diameter. The researcher designed hybrid interconnection networks built on two or more core network topologies to improve network topology [9]. Examples of hybrid interconnection networks:

Varietal hypercube (VH) interconnection network topology for big multicomputer computers. The hypercube offers recursive structure, partition ability, high connectivity, and the ability to integrate other architectures like ring and mesh. The network has the same number of nodes and links. The varietal hypercube is two-thirds the diameter of the hypercube. The varietal hypercube has a shorter average distance than the hypercube. The shortest path communication is guaranteed by optimal routing and broadcasting techniques [10].

The hex-cell (HC) is a new fundamental interconnection network. Its design decreases overall cost by requiring fewer connections. For large networks, its large diameter compared to a hypercube is negative [11].

Multilayer hex-cells (MLH) consider a unique hybrid interconnection network. Their construction is built on a hexagonal grid (HC) [12]. (MLH) has a larger diameter than the hypercube topology. However, it is extendable and better than (HC).

Chained-cubic tree (CCT) interconnection networks are built on tree and hypercube designs [4, 13]. This structure uses chains of hypercubes in a tree shape to take advantage of both structures. (CCT) balances the two architectures by keeping their strengths and limiting their drawbacks. Because of its narrow bisection breadth and lack of parallel paths, the tree topology is impractical despite the tree topology's good maximum node degree and adequate diameter when employed with a basic routing algorithm.

The tree-hypercube (TQ) has many hypercube properties, including self-routing and division [14]. They outperform hypercubes in diameter, extension, and average distance. It was also shown that the (TQ) has considerable division flexibility.

Hypercube and mesh interconnection networks are combined in the hyper-mesh (HM) network [15]. The (HM) networks use the hypercube network and merge it with the mesh network, which has a lower fixed degree and does not grow in value as the size network study shows that the (HM) has the desired characteristics of its two component networks, significantly mitigating the HM's fundamental defects.

In terms of speed, the mesh-of-trees (MOT) network is seen as the best choice because it has two good qualities: a small diameter and a wide bisection [16].

3. HYPERCUBE

The hypercube is one of the best significant connection networks for its topological features. These topological qualities include low diameter and high connectedness. An n-cube is an undirected network with 2^n nodes with labels from 0 to 2n -1. Degree (n) is a node's number of links. One kind of building known as logarithmic architecture is the hypercube. This is because $\log_2 N=n$ links are the maximum number of connections that a message must travel across to arrive at its destination within an n-cube [4, 17]. The diameter hypercube, n or $\log_2 N$, represents the most nodes a message must pass through to reach its mate. Fig. 2 illustrates several hypercube dimensions [12, 18].

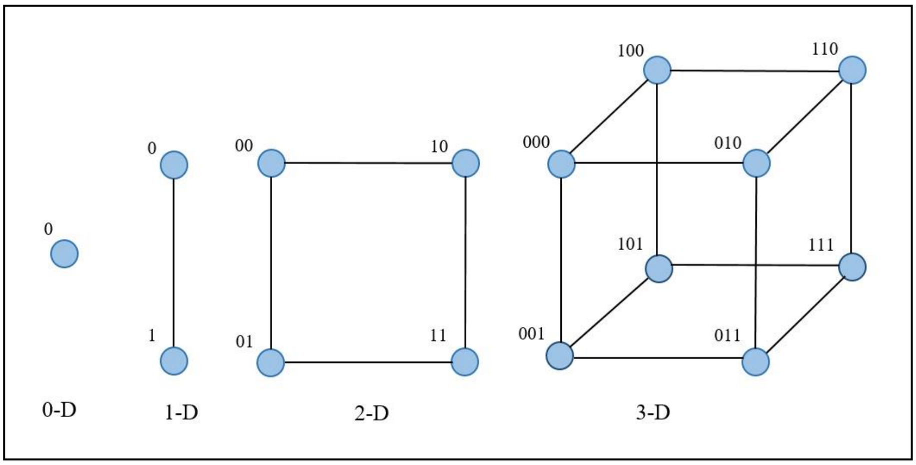

FIGURE 2. - Hypercube with varying dimensions and numbers of nodes [8]

The fact that hypercube networks are built using a recursive pattern is one of the many reasons why they are so popular. By linking nodes with comparable addresses in both subcubes, it t is feasible to construct an n-cube by first making two subcubes, each of which must have a degree value of (n - 1). Note that the 4-cube shown in Fig. 3 may be divided into two main subcubes with a degree of three. Note that to build the 4-cube out of the two 3-cubes, the degree of each node has to be increased [3, 6].

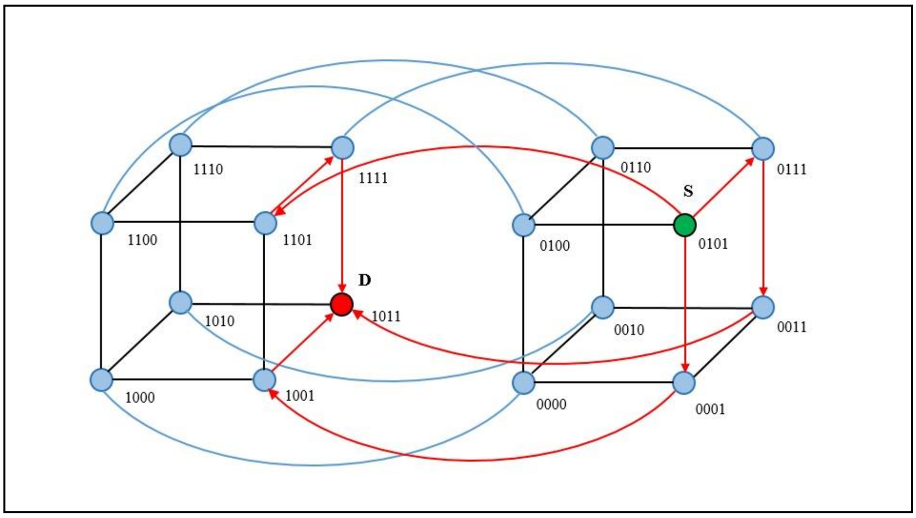

FIGURE 3. -. A four-dimension hypercube [8]

Each node's number is assigned, so there is a difference of one binary bit between it and the n nodes immediately adjacent to it and directly related to it. By XORing the binary address representations of nodes i and j, it is feasible to learn the path travelled by the transmission that began at node i and was meant for node j. If the XOR function produces one in a particular bit location, the message must be delivered with the appropriate connection. When used in preceding Fig. 3, the XOR operation yielded the value (1110) when a message was conveyed from the source (S) node (0101) to the destination (D) node (1011). Only dimensions 2, 3, and 4 (counting from right to left) will be used for the message's transmission [3, 8].

The evaluation of a topology of interconnection networks takes the place of numerous parameters. Table 1 covers a variety of different interconnection networks [15, 19].

Table 1. - Performance characteristic of static networks [8]

Network	Degree (n)	Diameter (dim)	Cost (No. of links)	Worst delay
CCNs	N-1	1	N(N-1)/2	1
Linear array	2	N-1	N-1	N
Binary tree	3	$2(\lceil \log_2 N \rceil - 1)$	N-1	$\log_2 N$
n-cube	$\log_2 N$	$\log_2 N$	nN/2	$\log_2 N$
2D-mesh	4	2(n-1)	2(N-n)	$\sqrt{N}$
k-ary n-cube	2n	$N\lfloor k/2 \rfloor$	nxN	$k\log_2 N$

4. PRINCIPLES OF FRACTALS

Fractals are patterns and formations constructed of geometric shapes that repeat their geometry at various scales, from very small to extremely large. Fractals can describe structures and surfaces that cannot be modeled using ordinary Euclidean geometry. Fractals are used extensively in science, technology, and art [20].

4.1 SIERPINSKITRIANGLE

The well-known Sierpinski triangle is constructed using an iterative approach. As seen in Fig. 4, for the first four iterations of the conventional Sierpinski triangle fractal, the basic unit of the Sierpinski triangle is the equilateral triangle [20, 21]. When level (*m*) equals zero, the triangle is centered at the origin, and (*l*) is used to determine the length of each of its sides. At the second level (*m=2*), the triangle is cut into four smaller triangles, each with an edge length of (*l/2*). The three triangles located at the corners of the original triangle are maintained, while the triangle in the center is eliminated, as shown in Figure. In the same way, the rest of the levels are generated. The equation that may be used to determine the number of triangles that make up the ST at any arbitrary iteration number m is as follows [22, 23]:

$$N_m = 3^m \quad (1)$$

Likewise, the side length of each triangle may be calculated using the following formula at the m-th iteration:

$$l_m = \frac{l}{2^m} \quad (2)$$

As a result, by using Equations (1) and (2), we may deduce the following about the fractal dimension:

$$D = \lim_{m \to \infty} \frac{\log N_m}{\log(l / l_m)} \approx 1.58 \quad (3)$$

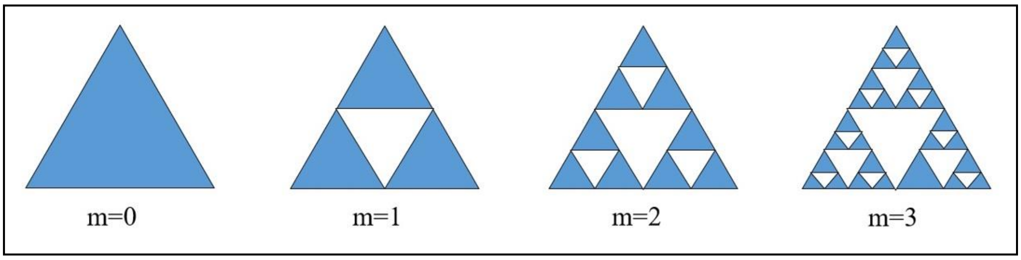

FIGURE 4. - The initial iteration four levels of Sierpinski triangle [20]

4.2 SIERPINSKI CARPET

The Sierpinski carpet is an example of an ideal fractal object analogous to the Sierpinski triangle [24, 25]. The carpet is distinctive in that it is made up of main particles that are square in shape. A fundamental square particle makes up the first level of the structure. For a level two aggregate, eight primary particles should be arranged in a square loop with a square hole the size of the primary particles in the middle. To create a level three aggregate, arrange eight level two totals in a square loop, leaving a hole in the center that is the same size as the level two accommodation (Fig. 5 demonstrates a three-level carpet) [25, 26].

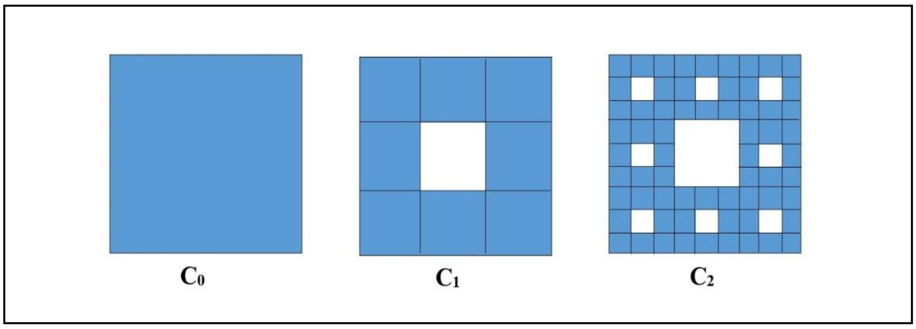

FIGURE 5. - The Sierpinski carpet during the initial three levels [24]

5. METHODOLOGY OF SUGGESTION FRACTAL TOPOLOGIES

In this section, two main fractal topologies for creating interconnection networks are presented, both based on the design of a hypercube interconnection network.

5.1 THE SIERPINSKI TRIANGLE TOPOLOGY (STT)

The proposed construction of the Sierpinski Triangle Topology (STT) is based on the fractal recursive formula Sierpinski Triangle described in (section IV.A). In the Sierpinski Triangle Topology (STT)-based multiprocessor system, processing elements are positioned at the graph's vertices. Edges of the graph represent the point-to-point communication links between processors. This architecture is created initially in a one-dimensional (D1) triangle with three nodes and three edges. Each node has a degree (n) of two, representing the number of links on the node (the number of neighbours nodes). The two-dimensional representation (D2), known as the Agent Sierpinski Triangle, comprises two triangles distinguished by six nodes. Noting that the three new nodes have a degree (n) of four. This Agent Sierpinski Triangle (D2) represents the critical step to build the base cell for the Sierpinski Triangle Topology (STT) to create a three-dimensional shape (D3) derived from it. It is mentioned that all of the processor nodes associated with this cell have a degree (n) of four, as seen in Fig. 6.

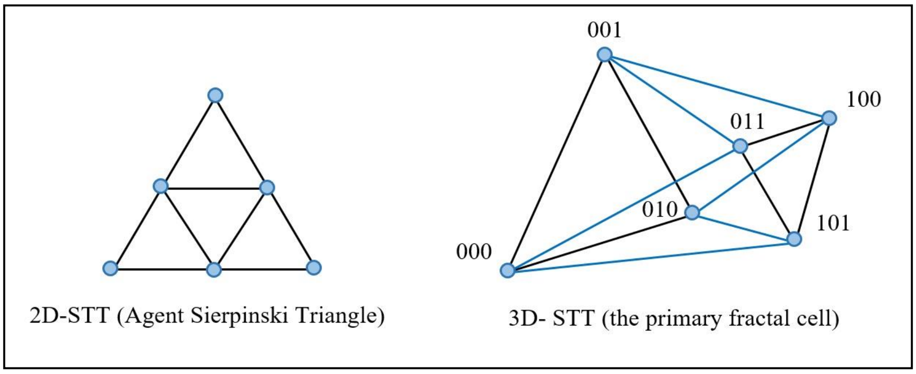

FIGURE 6. - The Sierpinski Triangle Topology (STT) 2D (Agent Sierpinski Triangle) and 3D (the primary fractal cell)

The three-dimensional (3D-STT) represents the primary fractal cell for the expansion of the interconnection network of this fractal topology, as is the case in the hypercube (D3). So that the distribution of the binary addresses of the nodes serves in a manner consistent with the application of the XOR-ing process, which is used to find paths. It is possible to create an n-Sierpinski Triangle Topology from two sub-Sierpinski Triangle Topologies, each of which has an (n -1) degree, by linking nodes with comparable addresses in both sets of sub-Sierpinski Triangle Topologies. For instance, the D4-Sierpinski Triangle Topology depicted in Fig. 7 builds by constructing two 3D-Sierpinski Triangle Topologies, each with a degree (n) of four.

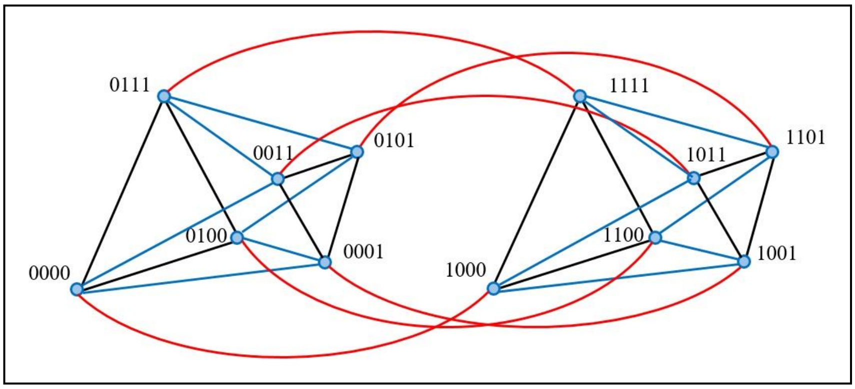

FIGURE 7. - A four-dimension of Sierpinski Triangle Topology (STT)

Message transfer will be tested via the proposed Sierpinski Triangle Topology (STT). If the sent is sent from the source (S) node 0001 to the destination (D) node 1111, the XOR operation will give the result 1110. For the message to reach its goal, it will have to be transmitted solely via dimensions 2, 3, and 4 (counting from right to left). Therefore, there are three distinct paths that the message might travel, and those paths are highlighted in red in bold in Fig. 8. So the set of three paths formed are:

$$\text{Path1}: 0001 \rightarrow 0011 \rightarrow 0111 \rightarrow 1111$$
$$\text{Path2}: 0001 \rightarrow 0101 \rightarrow 1101 \rightarrow 1111$$
$$\text{Path3}: 0001 \rightarrow 1001 \rightarrow 1011 \rightarrow 1111$$

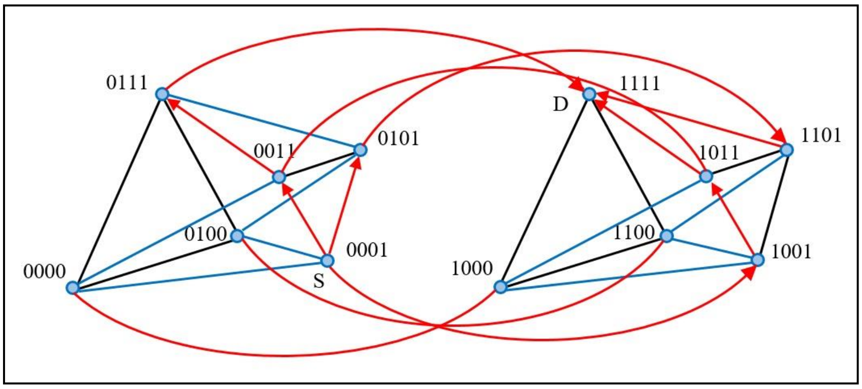

FIGURE 8. - Routing paths in the 4D-Sierpinski Triangle Topology (STT)

The algorithm that sends messages from the sender to the receiver can be described as follows:

Routing algorithm

Source node:

1. Specify source node (*S:* $s_1, s_2 \ldots s_n$) and destination node (*D:* $d_1, d_2, \ldots d_n$) in binary representation.
   // *n:* represents the degree of topology.
2. Procedure *XOR* operation $R = S \oplus D$
   // where *R:* $r_1, r_2 \ldots r_n$.
3. Specify the number of ones (k) in the (*R bits*), representing the number of paths and the number of hops per path.
   // where $0 \leq k \leq n$.
4. Determine paths (k).

Routing nodes:

1. For i = 1 to k do
2. Pass the message according to the path (p_i) specified in the step (4).
3. End.

The following is a description of the topological qualities that may be gained from Sierpinski Triangle Topology (STT):

1. The general formula for the number of processors or nodes in this topology is:

$$N = 3.2^{k} \tag{4}$$

for (D ≥ 1) and (k=0,1,2, 3...m).

2. The number of links (edges) is equivalent to:

$$L = \frac{N.n}{2} \tag{5}$$

for (D ≥3) and (n) is the degree (links per node).

3. The bisection distance is:

$$BD = 3.2^{n-4} \tag{6}$$

for (D ≥3).

5.2 THE SIERPINSKI CARPET TOPOLOGY (SCT)

The fractal recursive formula Sierpinski Carpet, detailed in (section IV.B), serves as the foundation for the proposed creation of the Sierpinski Carpet Topology (SCT). As mentioned, the development of this topological system is achieved by utilizing the concepts of the hypercube and the mesh as fundamental construction principles. In a multiprocessor system based on Sierpinski Carpet Topology (SCT), processing elements are placed at the graph's vertices. The graph's edges show how processors can communicate with each other directly.

The initial design starts from a square in a one-dimensional (D1), with four nodes (processors) and four edges. Every node in the network has a degree (n) of two, which denotes the number of links with the node. After that, it notices that the two-dimensional (D2), known as the Agent Sierpinski Carpet, is more like the Sierpinski carpet on the second level. This Agent Sierpinski Carpet (D2) represents the critical step to build the base cell for the Sierpinski Carpet Topology (SCT) to create a three-dimensional shape (D3) derived from it. When taking a closer look at the figure, each node has four links (n), except for the cross's terminal nodes (n-lower), which are colored red, where these nodes only have three links (n-1). Namely, this indicates that half of the total number of nodes has a degree of four, while the other half has only a degree of three, as shown in Fig. 9. The Topological properties can be summarized as follows:

The three-dimensional (3D-STC) represents the primary fractal cell for the expansion of the interconnection network of this fractal topology, as is the case in the hypercube (D3). So that the distribution of the binary addresses of the nodes serves in a manner consistent with the application of the XOR-ing process, which is used to find paths.

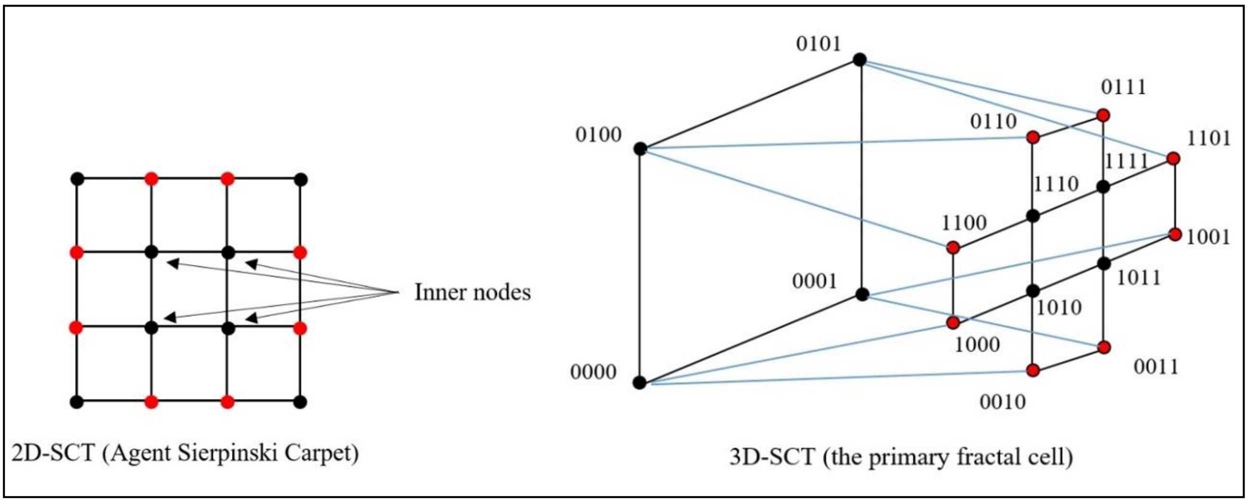

FIGURE 9. - The Sierpinski Carpet Topology (SCT) 2D (Agent Sierpinski Triangle) and 3D (the primary fractal cell)

It is possible to create an n-Sierpinski Carpet Topology from two sub-Sierpinski Carpet Topologies, each of which has an (n -1) degree, by linking nodes with comparable addresses in both sets of sub-Sierpinski Carpet Topologies. Note that the 4D-Sierpinski Carpet Topology depicted in Fig. 10 is built by constructing two sub-Sierpinski Carpet Topologies so that the degree of nodes of the built topology (4D) is incremented by one.

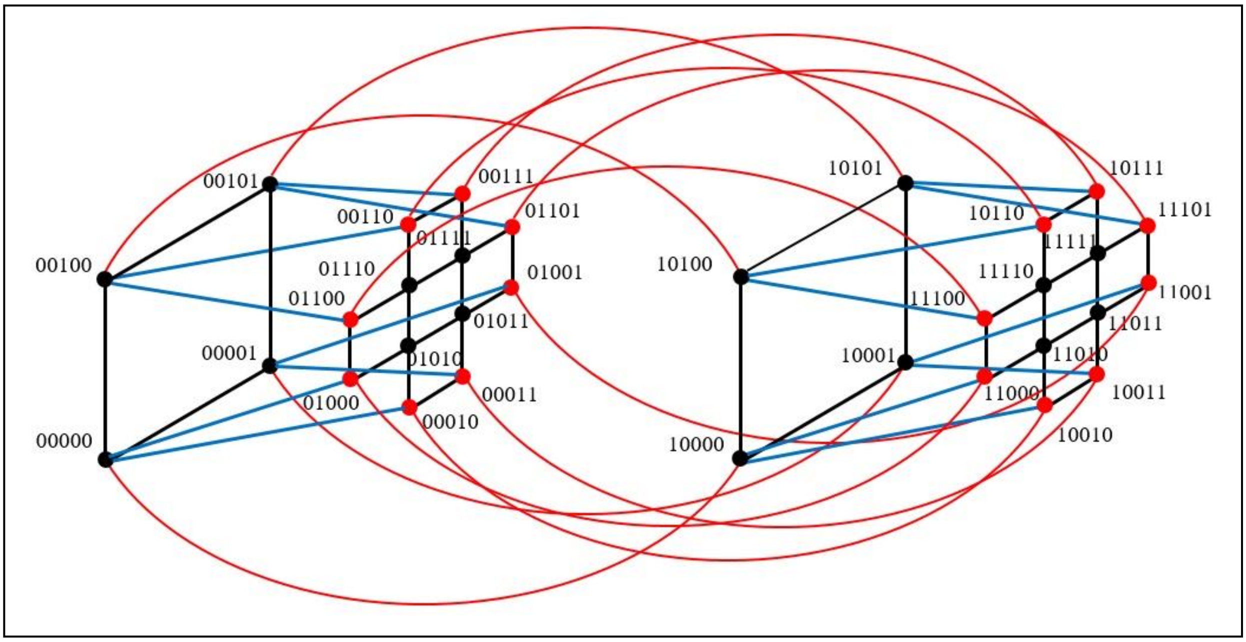

FIGURE 10. - A four-dimension of Sierpinski Carpet Topology (SCT)

Message transfer will be tested via the proposed Sierpinski Carpet Topology (SCT). If the sent is transferred from the source (S) node 00111 to the destination (D) node 10100, the XOR operation will provide the result 10011 in this scenario. For the message to reach its goal, it will have to be transmitted solely via dimensions 1, 2, and 5 (counting from right to left). So, there are three paths the message might travel, highlighted in red in bold in Fig. 11. So the set of three paths formed are:

$$\text{Path1}: 00111 \rightarrow 00110 \rightarrow 00100 \rightarrow 10100$$
$$\text{Path2}: 00111 \rightarrow 00101 \rightarrow 10101 \rightarrow 10100$$
$$\text{Path3}: 00111 \rightarrow 10111 \rightarrow 10110 \rightarrow 10100$$

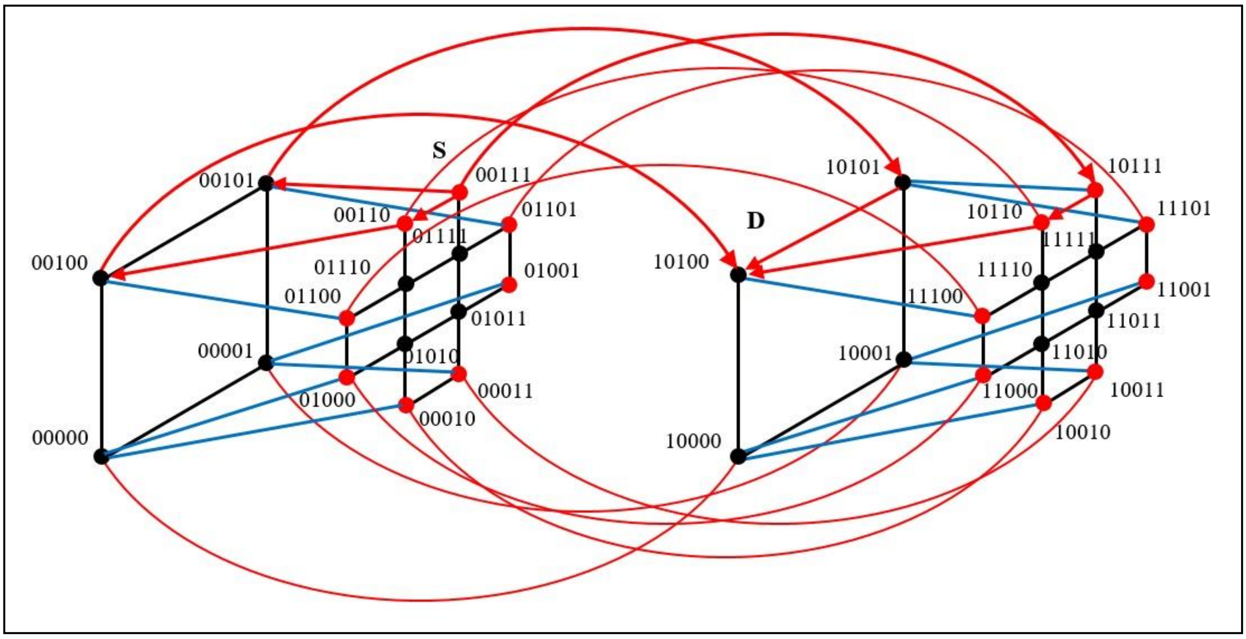

FIGURE 11. - **Routing paths in the 4D-Sierpinski Carpet Topology (SCT)**

The following is a description of the topological properties that may be gained from Sierpinski Carpet Topology (SCT):

4. The general formula for the number of processors or nodes in this topology for ($D \geq 3$) and ($k=1,2, 3...m$) is:

$$N = 8.2^k \quad (7)$$

5. The bisection distance for ($D \geq 4$) and ($k=2,3, 4...m$) is:

$$BD = \frac{N}{2} - 2^k \quad (8)$$

6. The number of links (L) for (D ≥4) and ($k=2,3, 4...m$) is equivalent to:

$$L_i = 2 * L_{i-1} + 3 * 2^k \quad (9)$$

For example, the number of links of 3D-Sierpinski Carpet Topology (L3) is 28 (see Fig. 9). Now it can calculate the number of links for 4D- Sierpinski Carpet Topology (L4) that has 32 nodes (N) from equation 9:

$$L_3 = 28$$
$$L4 = 2 * 28 + 3 * 2^2$$
$$L_4 = 68$$

And so, when expanding to the rest of the architectures for this topology.

In summary, Table 2 shows the obtained topological properties of these suggested interconnection networks with hypercubes.

Table 2. - **The topological characteristics of suggested topologies and hypercube**

Network	Diameter	N	Number of links	Bisection distance
n-cube	$\log_2 N$	2^n	N.n/2	2n-1
STT	$\log_2 N$	3.2^k	N.n/2	3.2^{n-4}
SCT	$\log_2 N$	8.2^k	$L_i = 2.L_{i-1} + \frac{N_i}{2}$	2^{n-1}

6. EXPERIMENT WORK

In this section, the routing algorithm of the proposed topologies is tested using the Capcarbon version 5 simulator and implemented on an HP Laptop (1.60GHz CPU, 20GB RAM).

Sierpinski Triangle Topology (STT) in Fig 8 tests by a Cupcarbon simulator. Fig. 12 illustrates that the nodes actively exchanging messages are responsible for most energy consumption, whilst the energy is not changed in the inactive nodes. Fig. 13 shows the calculated total consumption of the nodes.

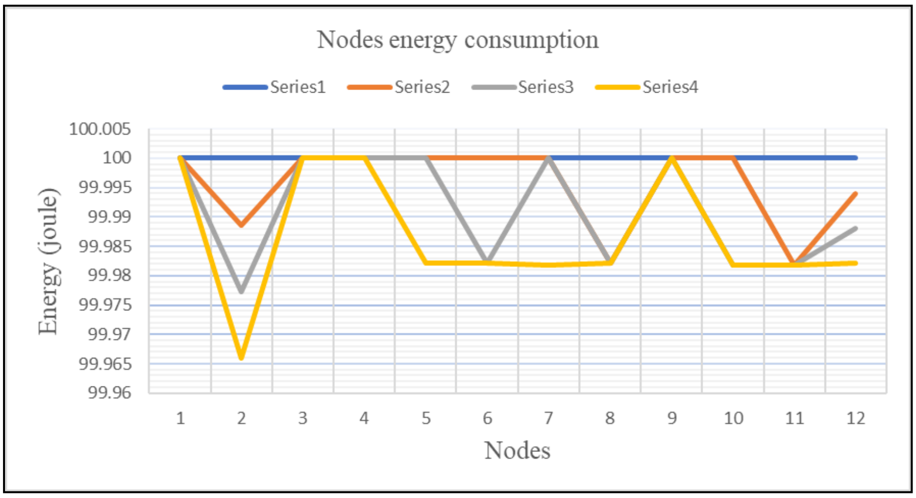

FIGURE 12. **- Energy consumption of nodes 4D- Sierpinski Triangle Topology (STT)**

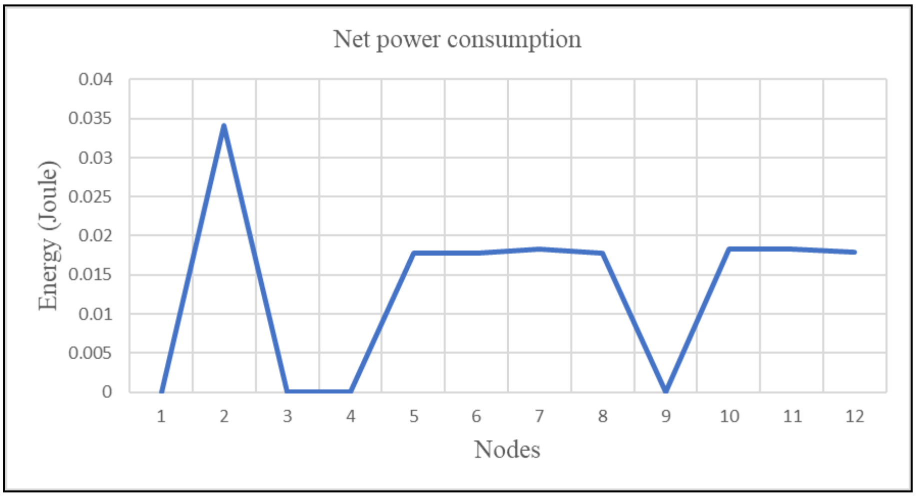

FIGURE 13. **-. Net energy consumption of nodes 4D- Sierpinski Triangle Topology (STT)**

Similarly, Sierpinski Carpet Topology (SCT) in Fig. 11 tests within a Cupcarbon simulator. Fig. 14 shows how much energy each node uses during the transmission process along each path. The computed aggregate of the nodes' total consumption is depicted in Fig. 15.

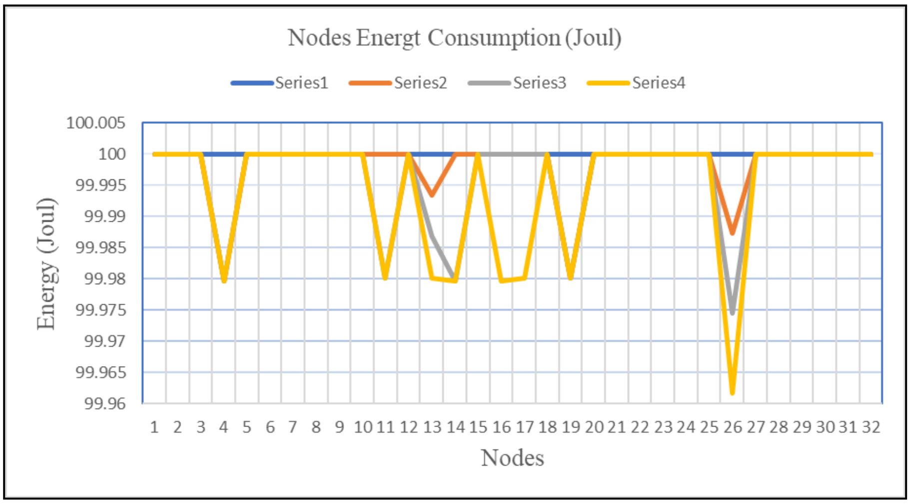

FIGURE 14. **- Energy consumption of nodes 4D-Sierpinski Carpet Topology (SCT)**

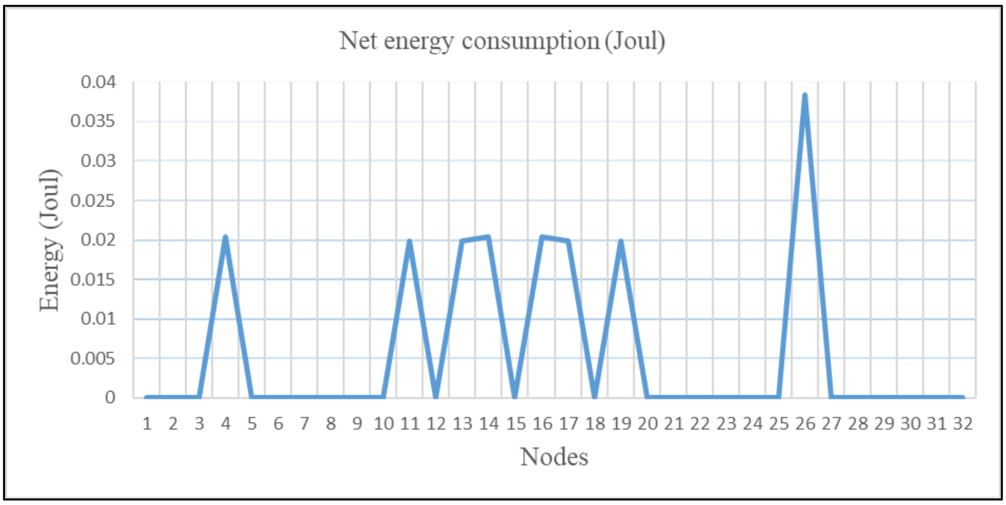

FIGURE 15. - Net energy consumption of nodes 4D-Sierpinski Carpet Topology (SCT)

7. CONCLUSION

Compared to a hypercube, the Sierpinski Triangle topology (STT) noticed a significant decrease in the number of nodes and links as large networks grew. This is an excellent way to lower costs because fewer nodes and links are needed. It also has shorter average distances and a more significant degree than a hypercube by one, which is better. However, this topology has a smaller bisection width and high diameter than a hypercube.

The Sierpinski Carpet Topology (SCT) has the advantage of having a high bisection width compared to a hypercube. That is preferable because it places a lower restriction on the difficulty of parallel algorithms by calculating the complexity based on the size of the data set divided by the bisection width in algorithms that need substantial amounts of data movement. But the drawback of this topology is that it has a diameter and average distance large than a hypercube.

FUNDING

No funding received for this work

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their efforts.

CONFLICTS OF INTEREST

The authors declare no conflict of interest

REFERENCES

[1] M. Abd-El-Barr and F. Gebali, “Reliability analysis and fault tolerance for hypercube multi-computer networks,” *Inf. Sci. (Ny).*, vol. 276, pp. 295–318, 2014.

[2] N. Phisutthangkoon and J. Werapun, “Shortest-path routing for optimal all-to-all personalized-exchange embedding on hierarchical hypercube networks,” *J. Parallel Distrib. Comput.*, vol. 150, pp. 139–154, 2021, doi: 10.1016/j.jpdc.2021.01.004.

[3] H. El-Rewini and M. Abd-El-Barr, *Advanced Computer Architecture and Parallel Processing*. 2005. doi: 10.1002/0471478385.

[4] B. A. Mahafzah, M. Alshraideh, L. Tahat, and N. Almasri, “Topological Properties Assessment for Hyper Hexa-Cell Interconnection Network,” vol. 13, pp. 115–121, 2019.

[5] F. Zhang, Y. Feng, and W. Feng, "Three-dimensional interconnected networks for thermally conductive polymer composites: Design, preparation, properties, and mechanisms," *Mater. Sci. Eng. R Reports*, vol. 142, p. 100580, 2020.

[6] F. Bistouni and M. Jahanshahi, "Impact of Raising Switching Stages on the Reliability of Interconnection Networks," *J. Inst. Electron. Comput.*, vol. 2, no. 1, pp. 93–120, 2020, doi: 10.33969/jiec.2020.21007.

[7] N. Gupta, K. S. Vaisla, and R. Kumar, "Design of a structured hypercube network chip topology model for energy efficiency in wireless sensor network using machine learning," *SN Comput. Sci.*, vol. 2, pp. 1–13, 2021.

[8] D. C. Marinescu, *Cloud computing: theory and practice*. Morgan Kaufmann, 2022.

[9] H. T. T. Binh, P. D. Thanh, and T. B. Thang, "New approach to solving the clustered shortest-path tree problem based on reducing the search space of evolutionary algorithm," *Knowledge-Based Syst.*, vol. 180, pp. 12–25, 2019.

[10] S.-Y. Cheng and J.-H. Chuang, "Varietal hypercube-a new interconnection network topology for large scale multicomputer," in *Proceedings of 1994 International Conference on Parallel and Distributed Systems*, 1994, pp. 703–708.

[11] M. Qatawneh, "Multilayer Hex-Cells: A New Class of Hex-Cell Interconnection Networks for Massively Parallel Systems," *Int. J. Commun. Netw. Syst. Sci.*, vol. 04, no. 11, pp. 704–708, 2011, doi: 10.4236/ijcns.2011.411086.

[12] B. A. Mahafzah, M. Alshraideh, T. M. Abu-Kabeer, E. F. Ahmad, and N. A. Hamad, "The Optical Chained-Cubic Tree interconnection network: Topological structure and properties," *Comput. Electr. Eng.*, vol. 38, no. 2, pp. 330–345, 2012, doi: 10.1016/j.compeleceng.2011.11.023.

[13] Y. Saad and M. H. Schultz, "Topological properties of hypercubes," *IEEE Trans. Comput.*, vol. 37, no. 7, pp. 867–872, 1988.

[14] N. Adhikari, A. Singh, and N. K. Swain, "Reliable, Effective and Fault-Tolerant design of Leafy cube interconnection network topology," *Int. J. Innov. Technol. Explor. Eng*, vol. 8, no. 12, pp. 3163–3170, 2019.

[15] S. Loucif, M. Ould-Khaoua, and A. Al-Ayyoub, "Hypermeshes: implementation and performance," *J. Syst. Archit.*, vol. 48, no. 1–3, pp. 37–47, 2002.

[16] W. M. Chen, G. H. Chen, and D. F. Hsu, "Generalized diameters of the mesh of trees," *Theory Comput. Syst.*, vol. 37, no. 4, pp. 547–556, 2004, doi: 10.1007/s00224-004-1115-0.

[17] A. Y. Zomaya, "Advanced Computer Architecture and Parallel Processing." 2021.

[18] A. I. Qatawneh, "Embedding Hex-Cells into Tree-Hypercube Networks," *Int. J. Comput. Sci. Issues*, vol. 10, no. 3, p. 136, 2013.

[19] W.-M. Chen, G.-H. Chen, and D. F. Hsu, "Generalized diameters of the mesh of trees," *Theory Comput. Syst.*, vol. 37, pp. 547–556, 2004.

[20] N. R. Mosteanu and A. Faccia, "Fintech frontiers in quantum computing, fractals, and blockchain distributed ledger: Paradigm shifts and open innovation," *J. Open Innov. Technol. Mark. Complex.*, vol. 7, no. 1, p. 19, 2021.

[21] A. Husain, M. N. Nanda, M. S. Chowdary, and M. Sajid, "Fractals: An Eclectic Survey, Part-I," *Fractal Fract.*, vol. 6, no. 2, p. 89, 2022.

[22] F. Jahanmiri and D. C. Parker, "An Overview of Fractal Geometry Applied to Urban Planning," *Land*, vol. 11, no. 4, 2022, doi: 10.3390/land11040475.

[23] Y. Li and others, "Average geodesic distance on stretched Sierpiński gasket," *Chaos, Solitons \& Fractals*, vol. 150, p. 111120, 2021.

[24] K. Kavitha, "Design of a Sierpinski Gasket Fractal Bowtie Antenna for Multiband Applications," *Int. J. Appl. Eng. Res.*, vol. 13, no. 9, pp. 6865–6869, 2018, [Online]. Available: http://www.ripublication.com

[25] X. Yang, W. Zhou, P. Zhao, and S. Yuan, "Confined electrons in effective plane fractals," *Phys. Rev. B*, vol. 102, no. 24, pp. 1–10, 2020, doi: 10.1103/PhysRevB.102.245425.

[26] E. Shamsi, M. A. Ahmadi-Pajouh, and T. S. Ala, "Higuchi fractal dimension: An efficient approach to detection of brain entrainment to theta binaural beats," *Biomed. Signal Process. Control*, vol. 68, p. 102580, 2021.