DOI: http://dx.doi.org/10.26483/ijarcs.v10i6.6483

Volume 10, No. 6, November-December 2019

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

TOPOLOGICAL PROPERTIES OF NETWORK AND INFORMATION FLOW FOR PARALLEL AND DISTRIBUTED SYSTEM

Rakesh Kumar Katare[1], Shrinivash Premikar[1], and Charvi K[2],
Department of Computer Science, APS University, Rewa (MP)[1,]
Jawaharlal Nehru College of Technology, Ratahra, Rewa (MP)[2]

Abstract: In this paper we are analyzing and developing algorithms for the connectivity and complexity of point to point network. For the development of communication network algorithms we will convert the point to point network into its connectivity matrix. These connectivity matrix relations will be depending on the Processors to Processors relation or processor to edge relation or edge to edge relations. The logical operations will be performed between the vectors of nodes or edges of the connectivity matrix. These operations will be used to explore the topological properties of the architecture for information flow.

Keywords: Interconnection Network, Point to Point network, Connectivity matrix algorithms.

INTRODUCTION

Let ***G = (v, e)*** is a graph where v is the set of vertexes of a point-to-point communication network and e is the set of edges. The information can be sent noiselessly from node ***i*** to node ***j*** (for all ***(i, j)*** ∈ *e)* through edges of this communication network.

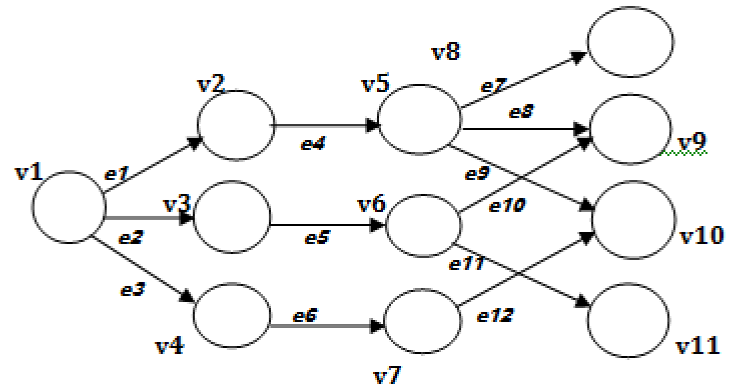

Figure: 1 Vertex-edge relation representing graph.

Now we are analyzing the connectivity and complexity of point to point network by converting the network into its equivalent connectivity matrix so that communication algorithm to connectivity can be developed. Here we will use vectors of the matrix for analyzing the connectivity of nodes. The logical operation between vectors will be used for the study of topological property of the architecture for information flow [1] [2].

Edges / Vertices	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12	Degree
v1	1	1	1	0	0	0	0	0	0	0	0	0	3
v2	1	0	0	1	0	0	0	0	0	0	0	0	2
v3	0	1	0	0	1	0	0	0	0	0	0	0	2
v4	0	0	1	0	0	1	0	0	0	0	0	0	2
v5	0	0	0	1	0	0	1	1	0	1	0	0	4
v6	0	0	0	0	1	0	0	0	1	0	1	0	3
v7	0	0	0	0	0	1	0	0	0	0	0	1	2
v8	0	0	0	0	0	0	1	0	0	0	0	0	4
v9	0	0	0	0	0	0	0	1	1	0	0	0	3
v10	0	0	0	0	0	0	0	0	0	1	0	1	4
v11	0	0	0	0	0	0	0	0	0	0	1	0	3
No of Vertexes (Edges value)	2	2	2	2	2	2	2	2	2	2	2	2	
	v1v2	v1v3	v1v4	v2v5	v3v6	v4v7	v5v8	v5v9	v6v9	v5v10	v6v10	v7v10	

Table.1 (a) vertex to edges relation of figure 1.

STRUCTURAL RELATION OF NODES AND VECTORS

Here we are representing the structural relation between nodes & edges of any graph by converting matrices. We are assuming that the relation between node to itself or edge to itself is possible and we will take 1 for self node/self edge in our connectivity matrix.

The table 1 gives us degree of each node and number of vertexes connected with each edge. The table 2 shows vertex to vertex relation of figure 1. In table 3 we have represented edge to edge & vertex to vertex relations of figure 2 and figure 3. Table 1(b) shows the logical operation of edges with its compliments to shows tautology or contradiction for the purpose to validate the connectivity.

		Logical operation of edge & compliment of each edge	
edges	Compliment of each edge	with "AND"	with "OR"
e1(11000000000)	*e1(00111111111)*	(00000000000)	(11111111111)
e2(10100000000)	*e2(01011111111)*	(00000000000)	(11111111111)
e3(10010000000)	*e3(01101111111)*	(00000000000)	(11111111111)
e4(01001000000)	*e4(10110111111)*	(00000000000)	(11111111111)
e5(00100100000)	*e5(11010011111)*	(00000000000)	(11111111111)
e6(00010010000)	*e6(11101101111)*	(00000000000)	(11111111111)
e7(00001001000)	*e7(11110110111)*	(00000000000)	(11111111111)
e8(00001000100)	*e8(11110111011)*	(00000000000)	(11111111111)
e9(00000100100)	*e9(11111011011)*	(00000000000)	(11111111111)
e10(00001000010)	*e10(11110111101)*	(00000000000)	(11111111111)
e11(00000100001)	*e11(11111011110)*	(00000000000)	(11111111111)
e12(00000010010)	*e12(11111101101)*	(00000000000)	(11111111111)

Table 1(b) Logical Operation of edge with its compliments.

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	Degree
v1	0	1	1	1	0	0	0	0	0	0	0	3
v2	0	0	0	0	1	0	0	0	0	0	0	1
v3	0	0	0	0	0	1	0	0	0	0	0	1
v4	0	0	0	0	0	0	1	0	0	0	0	1
v5	0	0	0	0	0	0	0	1	1	1	0	3
v6	0	0	0	0	0	0	0	0	1	0	1	2
v7	0	0	0	0	0	0	0	0	0	1	0	1
v8	0	0	0	0	0	0	0	0	0	0	0	0
v9	0	0	0	0	0	0	0	0	0	0	0	0
v10	0	0	0	0	0	0	0	0	0	0	0	0
v11	0	0	0	0	0	0	0	0	0	0	0	0

Table. 2 vertex to vertex relation of figure 1.

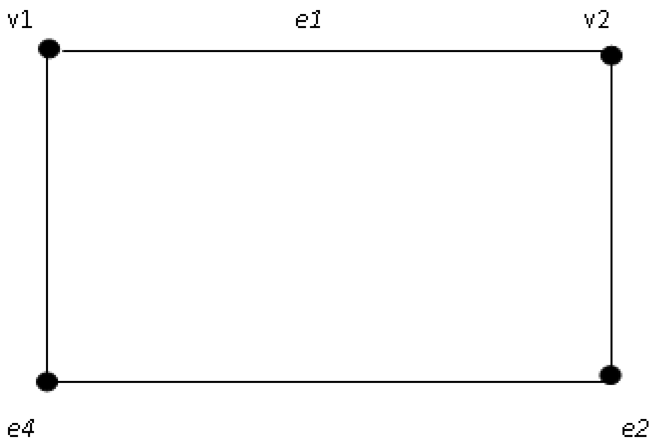

Fig.2 Graph Representation

	e1	e2	e3	e4
e1	1	1	0	1
e2	1	1	1	0
e3	0	1	1	1
e4	1	0	1	1

	v1	v2	v3	v4
v1	1	1	0	1
v2	1	1	1	0
v3	0	1	1	1
v4	1	0	1	1

Table.3 edge - edge & vertex – vertex Operation.

Table 5 shows the logical relation between processors of figure 3. In figure 3 we have two graph which are converted in vertex – vertex relations as tables shown in table3. Now we are using logical operation "AND" between vectors of matrix shown in table3. We have shown that "AND" gives common connectivity between two nodes.

$$v_i \ \textbf{AND} \ v_j = \begin{cases} 1 \text{ where } i=j=1 \\ 0 \text{ other wise} \end{cases}$$

The "AND" operation between compliments of nodes will give us contradiction where as "OR" operation between complimentary nodes gives tautology[3].

$$v_i \ \textbf{OR} \ v_j = \begin{cases} 0 \text{ if } i=j=1 \\ 1 \text{ other wise} \end{cases}$$

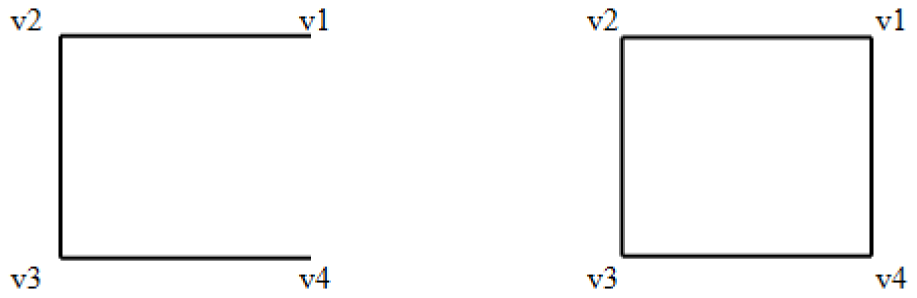

Fig. 3 Graph Representation

	v1	v2	v3	v4
v1	0	1	0	0
v2	1	0	1	0
v3	0	1	0	1
v4	0	0	1	0

(a)

	v1	v2	v3	v4
v1	0	1	0	1
v2	1	0	1	1
v3	0	1	0	1
v4	1	0	1	0

(b)

Table.4 (a) & (b) Matrix for vertex to vertex Operation.

Possible Connectivity of nodes with AND Operation:

Nodes	(a) AND	Possible Connectivity of nodes	(b)AND	Possible Connectivity of nodes
v1,v2	*v1(0 1 0 0) ^ v2(1 0 1 0) = {0 0 0 0}*	*Nil*	*v1(0 1 0 1) ^ v2(1 0 1 0) = {0 0 0 0}*	*Nil*
v1,v3	*v1(0 1 0 0) ^ v3(0 1 0 1) = {0 1 0 0}*	*(2)=1*	*v1(0 1 0 1) ^ v3(0 1 0 1) = {0 1 0 1}*	*(2,4)=2*
v1,v4	*v1(0 1 0 0) ^ v4(0 0 1 0) = {0 0 0 0}*	*Nil*	*v1(0 1 0 1) ^ v4(1 0 1 0) = {0 0 0 0}*	*Nil*
v2,v3	*v2(1 0 1 0) ^ v3(0 1 0 1) = {0 0 0 0}*	*Nil*	*v2(1 0 1 0) ^ v3(0 1 0 1) = {0 0 0 0}*	*Nil*
v2,v4	*v2(1 0 1 0) ^ v4(0 0 1 0) = {0 0 1 0}*	*(3)=1*	*v2(1 0 1 0) ^ v4(1 0 1 0) = {1 0 1 0}*	*(1,3)=2*
v3,v4	*v3(0 1 0 1) ^ v4(0 0 1 0) = {0 0 0 0}*	*Nil*	*v3(0 1 0 1) ^ v4(1 0 1 0) = {0 0 0 0}*	*Nil*

Table 5 Matrix relation with AND operation.

Possible Connectivity of Nodes with OR operation

Nodes	(a) OR	Possible Connectivity of nodes	(b)OR	Possible Connectivity of nodes
v1,v2	*v1(0 1 0 0)* ∨ *v2(1 0 1 0) = {1 1 1 0}*	*(1,2,3)=3*	*v1(0 1 0 1)* ∨ *v2(1 0 1 0) = {1 1 1 1}*	*(1,2,3,4)=4*
v1,v3	*v1(0 1 0 0)* ∨ *v3(0 1 0 1) = {0 1 0 1}*	*(2,4)=2*	*v1(0 1 0 1)* ∨ *v3(0 1 0 1) = {0 1 0 1}*	*(2,4)=2*
v1,v4	*v1(0 1 0 0)* ∨ *v4(0 0 1 0) = {0 1 1 0}*	*(2,3)=2*	*v1(0 1 0 1)* ∨ *v4(1 0 1 0) = {1 1 1 1}*	*(1,2,3,4)=4*
v2,v3	*v2(1 0 1 0)* ∨ *v3(0 1 0 1) = {1 1 1 1}*	*(1,2,3,4)=4*	*v2(1 0 1 0)* ∨ *v3(0 1 0 1) = {1 1 1 1}*	*(1,2,3,4)=4*
v2,v4	*v2(1 0 1 0)* ∨ *v4(0 0 1 0) = {1 0 1 0}*	*(1,3)=2*	*v2(1 0 1 0)* ∨ *v4(1 0 1 0) = {1 0 1 0}*	*(1,3)=2*
v3,v4	*v3(0 1 0 1)* ∨ *v4(0 0 1 0) = {0 1 1 1}*	*(2,3,4)=3*	*v3(0 1 0 1)* ∨ *v4(1 0 1 0) = {1 1 1 1}*	*(1,2,3,4)=4*

Table 6 Matrix relation with OR operation.

CONCLUSION AND FUTUREWORK

In this paper we are trying to establish the connectivity and complexity of nodes for the development of algorithms. The following are the conclusion of vector operations between nodes of connectivity matrix.

1. The logical operation "AND" shows the connectivity of nodes in "Race" condition.
2. 2. The logical operation "OR" shows the connectivity of nodes when any of the node is active.
3. The complimentary nodes with "AND" not possible because of contradiction where as complements nodes with is always possible because of tautology.
4. The use of logical operations shows the validity of possible communication between nodes. The algorithms will validate the communication is possible or not at some point. We will show that binary relations are always possible in our future studies.

REFERENCES

[1] N. Deo, Graph Theory with Applications to Engineering and Computer Science.

[2] Ellis Horowitz and Sartaj Sahni, *Fundamentals of Data Structures and Fundamental of Data Structure in C*, Computer Science.

[3] Katare R K and Chaudhary N S, “Study of topological property of interconnection networks and its mapping to Sparse Matrix model” International journal

[**4**] I. Foster, C. Kesselman, and S. Tuecke, “The Anatomy of the Grid: Enabling Scalable Virtual Organizations,” Int. J. of High Performance Computing Applications, Vol. 15, No. 3, 2001, pp. 200 - 222.

[5] B. Bollobas, 1979 *Graph Theory, An Introductory Course*. New York: Springer-Verlag.