Annales UMCS Informatica AI 5 (2006) 379-391

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

The computer network topologies designs and the coherent graphs algebra

Sergiy L. Krivoi, Mirosław Hajder, Paweł Dymora, Mirosław Mazurek*

Department of Distributed Systems, Faculty of Electrical and Computer Engineering, Rzeszów University of Technology, Wincentego Pola 2, 35-959 Rzeszów, Poland

Abstract

This work presents one of the regular graphs algebra possible variants and its application in designing and analysis of computer network topologies characteristics, focused on fault tolerance of computer networks.

1. Introduction

This work is a continuation of surveys presented in papers [1,2]. The proposed regular graphs algebra came into being according to tasks of designing the computer networks (CN) topologies and an analysis of their characteristics from the fault tolerance of computer network links and nodes point of view. The CN is modeled as an undirected graph with the nodes corresponding to computers and edges – corresponding to connection links between computers. Selection of one or another CN topology is related with determining the topology fault tolerance degree in the case of CN connection links or nodes failure. If correctly designed topology is an undirected coherent graph then a coherency degree of such a graph may be used as a CN fault tolerance measure. In this end, this paper presents a variant of a regular graphs algebra by means of which the CN fault tolerance degree is determined and also the CN topology from these algebra operations point of view.

2. Preliminary information

The graph $G(V,E)$ is called an undirected graph, where V – the number of graph nodes; E – the number of graph edges, which consists of unordered pairs of elements from V (where pairs (u,v) and (v,u) are considered as the same).

*Corresponding authors: *e-mail addresses*: {krivoi; mhajder; dymorap; mirekmaz}@prz.rzeszow.pl

Graph G is called a finite graph if its nodes number V is finite. If $(u,v) \in E$, then nodes u, v are called the ends of this edge. The edge $e \in E$ is called incident to node $u \in V$, if this node is the edge e end. The node $u \in V$ degree is called a number of edges incident to this node. The node u degree is denoted as $n(u)$. If a node degree is equal zero, then such a node is called isolated and if a node degree is equal one then such a node is called ending or hanging [1,2].

In this work as a graph, the finite undirected graph is meant.

Let the graph $G(V,E)$ and two nodes $u,v \in V$ of this graph, are given. It is said, that nodes u and v are connected between one another with a path in a graph, if there exists intermediate nodes $u_1, u_2, ..., u_k$ of the graph G such that $u = u_1$, $v = u_k$ and for all $i = 1,2,...,k-1$ the $(u_i, u_{i+1}) \in E$ respectively. The number k – 1 is called a length of this path. The path, where the first and the last node covers, is called a cycle. The graph, where no cycle exists is called an acyclic graph [3].

Definition 1. *Graph G(V,E) is called coherent, if any two nodes of this are connected between one another with a path. Coherent acyclic graph is called a tree.*

A tree node is called internal if its node degree is greater than one (otherwise the node is an ending node) [4].

If between nodes u and v there exists a path with the length $k-1$, then in particular if $k=1$ then the path length equals 0. It means that $u = u_1 = u_k = v$, i.e. that from node u to node u there exists a path of length 0. As follows from this observation the relation R describing a "coherent path between itself" is reflexive, secondly, in the case of an undirected graph this relation is symmetric and of course transitive. It means this is the equivalency relation and there exists a quotient set $V/R = \{V_1, V_2, ..., V_m\}$, where V_i is an equivalency class of this relation such that $V_i \cap V_j = \varnothing$ where $i \neq j, i, j = 1,2,...,m$.

From this simple observation the below theorem, follows.

Theorem 1. *Graph G(V,E) is coherent if and only if when V/R = {V}, where R – the above introduced equivalency relation.*

Proof of the above theorem directly results from the definition and properties of the relation R.

From this theorem there follows a simple and efficient algorithm for determining the graph $G(V,E)$ coherency. In reality, to this end from any chosen node $u \in V$ we should walk around all nodes of the graph G and denote all nodes which have already been visited. If walking around is done and if any node is left indented then the graph is incoherent, otherwise coherent. It is obvious that in order to do it a depth search algorithm is used (DFS – Depth First Search).

Depth search algorithm is widely presented in book [4] thus it is not presented in this work.

Depth search algorithm turns out to be very important in analyzing the graphs coherency and its importance increases if in coherent graph or graphs there are some operations performed. Some of these operations may result in graphs incoherency and then the correct result will not be received. Obligatory, the operations must be checked from its performance influence point of view on network coherency [3,5].

3. The coherency degree of a coherent graph

3.1. The edge removal operation in a coherent graph

Let us remind that an edge $e=(u,v)$ removal operation in a graph $G=(V,E)$ leads to the graph $G-e=(V,E\setminus\{(u,v)\})$. Let us consider one question, how many edges may be removed from the coherent graph $G=(V,E)$ in order that the realization of these operations would not cause the state in which the connection graph is incoherent. Let us introduce the following definition [4].

Definition 2. *The maximal number d of graph edges whose removal from a graph does not cause changing of the graph coherency class whereas removing of d* + 1 *-th edge causes the graph incoherence, is called a coherency degree of the coherent graph.*

It is obvious that the graph obtained as a result of applying the edge removal operation in the coherent graph, is coherent if this edge belongs to one of its cycles. As a consequence, the following theorem is obtained [3].

Theorem 2. *The coherency degree d of the coherent graph* $G=(V,E)$ *is equal to its cyclic number* $|E|-|V|+1$.

Proof of the above theorem results from the fact that a coherent graph stays coherent if not more than $|E|-|V|+1$ edges are removed. Edge removal from such a graph results in obtaining a tree of the graph *G*. However, even if one edge in a tree is removed it causes the graph incoherency.

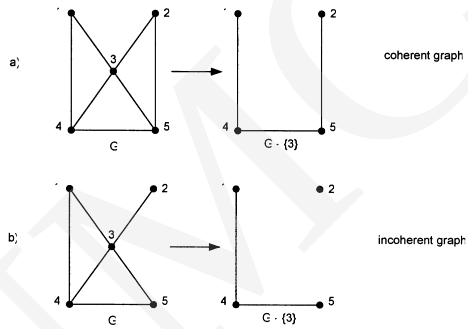
Conclusion 1. *Number of possible edge removal operations in a coherent graph* $G(V,E)$ *not causing the change of a graph coherency is equal to its cyclic number* $d=|E|-|V|+1$.

3.2. The node removal operation in a coherent graph

The node *u* removal operation from a graph $G=(V,E)$ is simplified to removing the node *u* from the nodes set *V*, and removing from an edge set *E* all incident edges to the node u. Directly from the above theorem the following theorem is obtained [4].

Theorem 3. *Graph, received as a result of applied node removal operation in a coherent graph is still coherent, if all incident edges to this node belong to some of its cycles or this node is an end node.*

Example 1. Let us consider the graph G, from which the node number 3 is removed.



In the case a) all edges incident to node 3, i.e. (1,3),(2,3),(4,3),(5,3) belongs to the cycles (1,3,4),(2,3,5),(4,3,5),(5,3,2),(5,3,4).

In the case b) the edge $(2,3)$ does not belong to any cycle of this graph.

In the conclusion of the previous theorem, the following dependencies occur:

– if $n(u) = 1$ the coherency degrees of graphs G and $G-u$ are equal, where u is a node of a graph G,
– the coherency degree of a graph obtained after applying m node removal operations from a coherent graph $G = (V,E)$ equals:

$$d' = d - (k_1 - 1) - \ldots - (k_m - 1),$$

where: d – the coherency degree of a graph G, k_i – the node u_i degree, u_i – a node removed from a graph where $i = 1,2,\ldots,m$.

4. The finite coherent undirected graphs algebra operations

In order to build the coherent graphs algebra it is necessary to determine the operations with respect to those whose set of all finite coherent undirected graphs is finite. It means that if as arguments of these operations the coherent graphs are taken then as a result of applying these operations the coherent graphs are also obtained.

If such algebra is built in the first step then in the second step it is necessary to examine properties of such algebra operations and to set its identity and type.

Let us notice that depending on applying the algebra operations set may vary. Because of that in this paper only one of such algebra possible variants is presented, which is focused on analysis of CN fault tolerance [1,2].

4.1. Fully specific operations

Let coherent graphs $G = (V_1,E_1)$ and $G_2 = (V_2,E_2)$ be given. Let us consider the operations for which the coherent graphs set is finite. Appropriate operations and principles of applying follows from the below theorems:

Theorem 4. *If for the coherent graphs the following operations are applied*

a) a bijective join (an isomorphic join operation),
b) a join operation,
c) the Cartesian Product operation,
d) an edge add operation,
e) a node add operation,

then as a result a coherent graph is obtained.

Proof.

a) The bijective join operation joins only two nodes u and $f(u)$, where f is *a bijective operation.* Naturally, it is enough to get a coherent graph coherent, thus in the obtained graph all nodes are connected with a certain path.
b) Because each node of the first graph is connected by a link with each node of the second graph, thus it is obvious that the property of coherency is preserved in a received graph.

Let us notice that bijective mapping existing in an isomorphic joint operation may be, in particular, graph isomorphism. If this bijection is isomorphism, thus the bijective join is called an isomorphic join.

The proof of the other theorems is obvious.

4.2. Partial operations

The operations which belong to this group may come from the coherent graphs class, thus these operations are partial. To this group of operations an edge and node removal operation belongs. These two operations need some explanations.

Theorem 5.

a) The graph sum $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$ *is a coherent graph if these graphs have at least one common node, i.e.* $|V_1 \cap V_2| > 0$.

b) *Graph obtained as a result of coherent graphs intersection is coherent, if as a result of applying graph nodes removal operation* $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$, *not belonging to the set* $V_1 \cap V_2$, *all edges incident to this nodes belong to some cycles of this graph or these nodes are incident with the edge ends.*

Proof.

a) *If* $v \in V_1 \cap V_2$ *then towards graphs* G_1 *and* G_2 *coherence it is received that from any node u of the graph* G_1 *exists a path to the node v in a graph* G_2*, and from node v exists a path to any node v' in a graph* G_2*. Then, from node* u *exists a path to node v', crossing a node v.*

b) *Proof follows from the definition of node removal operation. The result of this operation is not specified if as a result of applying this operation in the next step it leads to an incoherent graph. Besides, it is obvious that a graph intersection operation may be expressed by the operations of a node removal and a sum.*

Simple and sufficient condition of not applying the node removal operation is its negative meaning of the graph coherency degree which is received in the next step. Generally, in the case of ending of all node removal operations it is essential to apply the DFS algorithm for checking the coherency of a received graph. Its computing complexity, as known, is in proportion to the value of $O(\max(|E|,|V|)$.

5. Algebra definition

From the above considerations the finite coherent undirected graphs algebra follows. This algebra is partial.

Definition 3. *Pair AG = (A,Ω) is called the finite coherent undirected graphs algebra, if A – the algebra carrier – consists only of finite coherent undirected graphs, Ω – the algebra signature – includes operations such as: a sum, an intersection, the Cartesian product, a bijective join, a join, an edge and node input operation, an edge and node removal operation.*

Let us introduce a notation for these operations: $\cup^2, \cap^2, \times, *_f^2, *^2, w_{ik}^3, w_{iw}^3, w_{ok}^3$, w_{ow}^2 where the upper indices represent arity respectively to each operation.

5.1. Properties of the graph algebra

At the beginning, let us prove the theorem below.

Theorem 6. *The algebra $AG = (A,\Omega)$ as regards the sum and the intersection operations is a mesh.*

In order to prove genuineness of the principles we have to show idempotency of operations of the intersection and the sum, the commutation and unity, as well as right of absorption.

Let $G = (V,E)$ *and* $G_1 = (V_1,E_1)$ – any graphs.

Idempotency: at the basics of algebra we have:

a) $$G \cup G = (V \cup V, E \cup E) = (V, E) = G .$$

b) $$G \cap G = (V \cap V, E \cap E) = (V, E) = G .$$

The commutation and the unity of operations result directly from specifications of operations of the intersection and unity of graphs and truthfulness of analogous rights in algebra of multiplicity.

Absorption. At the basis of analogous rules of algebra of sets we have*:*

a) $$G \cup (G \cap G_1) = (V \cup (V \cap V_1), E \cup (E \cap E_1)) = (V, E) = G .$$

b) $$G \cap (G \cup G_1) = (V \cap (V \cup V_1), E \cap (E \cup E_1)) = (V, E) = G$$

As a consequence of the statement that all parts (elements, constituent parts) of the algebra of graphs presented above determine partly ordered multiplicity (ordered sets). Let us explain the order.

The order on the grid is being determined in the following way:

$G1 \le G2 \Leftrightarrow G1 \cap G2 = G1$ or with the second method $G1 \le G2 \Leftrightarrow G1 \cup G2 = G2$ but that means that $G1 \cap G2 = (V1 \cap V2, E1 \cap E2) = (V1, V2) = G1$ and as a result of the equivalence of conditions $A \subseteq B$ and $A \cap B = A$ we obtained $G1 \le G2 \Leftrightarrow G1 \subseteq G2$.

Now we will explain components which are minimal in this algebra (it is obvious that such components should exist). Existence of such components (type, kind, appearance) gives.

Conclusion 2. The minimal components of AG = (A,Ω) algebra are the trees in the form of T_u^0, where $T_u^0 = (V = \{u\}, E = \phi)$ – empty tree consisting of only one node at *u*.

Proof. Certainly trees play a particular part in this algebra since any coherent graph is built of trees. It means that the trees are generative in AG algebra. However, if we apply operations of removing the utmost node in regard to the tree, it will be reduced to the tree T_u^0. That is why empty trees play the role of minimal elements in this algebra. It is also obvious that empty trees consisting of only one node will not be reduced.

Taking advantage of this conclusion we will introduce into the analysis trees in the form of:

a) $$T_{u,v} = \left(V = \{u,v\}, E = \{(u,v)\}\right),$$

b) $$T_{u,v}^{w} = \left(V = \{u,w,v\}, E = \{(u,w),(w,v)\}\right).$$

These trees are being introduced by minimal members by means of the operation of AG algebra in the following way:

a) $$T_{u,v} = \left(V = \{u,v\}, E = \{(u,v)\}\right) = w_{ik}\left(T_u^0 \cup T_v^0, u, v\right) = \\ = T_{v,u} = T_u^0 * T_v^0 = T_u^0 *{}_f T_v^0\left(f(u) = v\right)$$

b) $$T_{u,v}^{w} = \left(V = \{u,w,v\}, E = \{(u,w),(w,v)\}\right) = \\ = T_{u,w} \cup T_{w,v} = w_{ik}\left(T_u^0 \cup T_w^0, u, w\right) \cup w_{ik}\left(T_w^0 \cup T_v^0, w, v\right) = \\ = \left(T_u^o * T_w^0\right) \cup \left(T_v^0 * T_w^0\right)$$

Let us also notice that the operation of the combination of two graphs is giving in effect the coherent graph even if one of the graphs is not coherent or both of them are so. In particular graphs corresponding to the expressions: $\left(T_u^0 \cup T_v^0\right) * G$ and $\left(T_u^0 \cup T_v^0\right) * \left(T_w^0 \cup T_s^0\right)$, where G – the coherent graph and *u, v, w, s* – the pairs of various tops will be coherent.

5.2. Dependence of the operation

Now we are testing the problem – which operations are basic, and which are not basic (i.e. these operations which are being expressed through basics).

1) The *operation of add of the node into the edge*. Let $G = (V, E) \in AG$ and $e = (u,v) \in E$. The operation of the insertion of the node w into the edge $e = (u,v)$ will transform the edge into 2 edges $e_1 = (u,w)$ and $e_2 = (w,v)$ in the graph G with removing the edge $e = (u,v)$. It is possible to realize this operation with the method:
$$w_{iw}\left(G, e, w\right) = \left(G \cup T_{u,v}^{w}\right) - e = w_{ok}\left(G \cup T_{u,v}^{w}, u, v\right).$$

Correctness of such a notation result from the fact that combination of two coherent graphs having common nodes, will also be a coherent graph but the edge e in the graph $G \cup T_{u,v}^{w}$ belongs to the *u, w, v, u* series. And therefore removing it from the graph will not infringe its coherence.

2) *The operation of add of the edge into the* coherent *graph G = (V,E)* is included in the fact that an edge is being introduced into the graph G between two not bordering (not encountering) nodes *u* and *v*. This operation, like the

previous one, is being expressed with the combination of the graph G and the tree $T_{u,v}$. Formally it is put down in the following way:

$$w_{ik}\left(G,u,v\right)=\left(G\cup T_{u,v}\right)=\left(G\cup T_{v,u}\right)=w_{ik}\left(G,v,u\right).$$

Correctness of such a notation result from the fact, that combination of two coherent graphs having common nodes, will also be a coherent graph. It is clear that also right is the identity in the form:

$$w_{ok}\left(w_{ik}\left(G,u,v\right),u,v\right)=w_{ik}\left(w_{ok}\left(G,u,v\right),u,v\right)=G.$$

3) *The operation of the combination of two coherent graphs $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$* where $V_1\cap V_2=\phi$, is presented in the obvious way through operations of the combination of graphs and sequences from $|V_1|*|V_2|$ of the operation of the insertion of the edge. This expression has the form:

$$G_1*G_2=\bigcup_{u\in V_1}\bigcup_{v\in V_2}w_{ik}\left(G_1\cup G_2,u,v\right)=\bigcup_{u\in V_1}\bigcup_{v\in V_2}\left(\left(G_1\cup G_2\right)\cup T_{u,v}\right).$$

Correctness of such law result from the fact that linking 2 nodes of coherent graphs with even one edge will result in receiving a coherent graph. It is obvious that the given operation is commutating and uniting.

4) *The operations of mutually unique combination of two coherent graphs $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$* depend on the connection with the corresponding nodes with mutually unique map. This operation is being expressed through operations of the combination of graphs and the sequence of the operation of the insertion of the edge. The expression is following for this form is:

$$G_1*_fG_2=\bigcup_{u\in V_1}\bigcup_{f(u)\in V_2}w_{ik}\left(G_1\cup G_2,u,f(u)\right)=\bigcup_{u\in V_1}\bigcup_{f(u)\in V_2}\left(\left(G_1\cup G_2\right)\cup T_{u,f(u)}\right).$$

Correctness of such a term, as in the previous case results from the fact that having linked 2 nodes of coherent graphs to at least one edge a coherent graph will be obtained. This and previous operations are commutating and uniting.

5) *The operation of the intersection of two coherent graphs $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$ depends on removing nodes in the graph $G_1\cup G_2$*, which are not common for these graphs. The expression for this activity is as follows:

$$G_1\cap G_2=\bigcup_{u\notin V_1\cap V_2}w_{ow}\left(G_1\cup G_2,u\right).$$

When drawing a conclusion let us notice that it is possible to show dependence on the operation of the Cartesian Product from other operations of this algebra, however, we will stop here and the algebra of coherent, ended not-directed graphs is assumes the form:

$$AG=\left(A,\Omega\right)=\left(A,\left\{\cup,\cap,\times,w_{ow},w_{ok}\right\}\right).$$

We will apply this algebra to examine various SK topologies and we will characterize them.

6. Characteristics of various SK topology

Let us consider some graphs often utilized to the construction of SK topology (look [3]) and we will characterize this topology of AG algebra determined from this algebra. We will introduce the term which will be utilized for the analysis of SK topology singly on this purpose.

Definition 4. We call the nodes of the coherent graph the point of the connection if removing this node from the graph lead to the incoherent graph.

Certainly any internal node is the point of the connection in the tree.

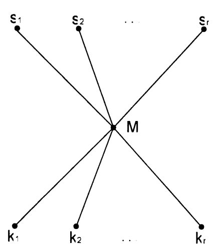
This definition results in the obvious way that the certain node will be a connection point to the point if he is adjacent to the final node. As follows from this simple remark the result of removing a node in the coherent graph will be indeterminate if this node is the point of the connection.

1) *Single-channel bus (tree).* In order to present SK with this topology we will consider the bus as the computer, representing the node of the graph. The graph lying at the basis of this topology has the form of the tree, shown below in Fig.1, where $s_1, s_2, \ldots, s_r$ – servers, $k_1, k_2, \ldots, k_n$ – computers, M – bus.

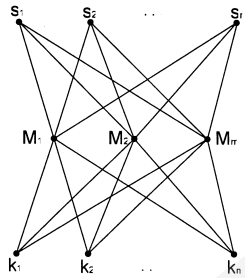
The given graph is the tree and it has 1 internal node (bus) with the degree of $n(M) = r+n$ and $r+n$ of terminal nodes. The algebraic expression in AG algebra has the form:

$$(T^0_{S1} \cup \ldots \cup T^0_{Sr} \cup T^0_{k1} \cup \ldots \cup T^0_{kn}) * T^0_M .$$

As we can see the topology is resistant to removing any number of terminal nodes and non-resistant in regard to removing the internal node (of bus) and removing at least one edge. Usage of reliable lines of the contact and the reliable bus is necessary for the usage of this topology.



2) *Multichannel bus (tripartite graph).* Improvement of the single-channel bus in the sense of raising SK reliability is a multichannel bus. The graph lying at the basis of this topology is the tripartite graph (Fig. 2), where $s_1, s_2, \ldots, s_r$ – servers, $k_1, k_2, \ldots, k_n$ – computers, $M_1, \ldots, M_m$ – buses.



The degree of cohesion of the graph equals $d = |E|\text{-}|V|+1 = rm+nm\text{-}r\text{-}n\text{-}m+1 = (r+n\text{-}1)(m\text{-}1)$. If we compare this topology with the previous one (in the previous topology m = 1 and d = 0), the degree of reliability increases m-1 times for this SK. The algebraical notation for this topology:

$$\left(T_{S1}^{0} \cup \ldots \cup T_{Sr}^{0}\right) * \left(T_{M1}^{0} \cup \ldots \cup T_{Mm}^{0}\right) \cup \left(T_{M1}^{0} \cup \ldots \cup T_{Mm}^{0}\right) * \left(T_{k1}^{0} \cup \ldots \cup T_{kn}^{0}\right) =$$

$$= \left(T_{S1}^{0} \cup \ldots \cup T_{Sr}^{0} \cup T_{k1}^{0} \cup \ldots \cup T_{kn}^{0}\right) * \left(T_{M1}^{0} \cup \ldots \cup T_{Mn}^{0}\right).$$

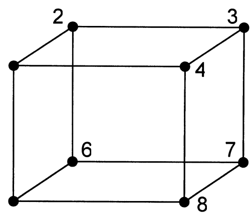
As we can see, falling out any m-1 buses leaves SK working (in this exception SK reduces to the single-channel bus, it is time to compare the notation with that of single-channel bus). Analogical statement refers to the edges.

3) *Cube.* The graph lying at the basis of this topology is a cube (Fig.3).

The degree of cohesion of the graph equals d = 12-8+1 = 5. It is possible for this graph to construct the algebraic expression with two methods:

a) $$\left(T_{1,2} \cup T_{2,3} \cup T_{3,4} \cup T_{4,1}\right) *_{f} \left(T_{5,6} \cup T_{6,7} \cup T_{7,8} \cup T_{8,5}\right),$$

b) $$\left(T_{1,2} \cup T_{2,3} \cup T_{3,4} \cup T_{4,1}\right) \times T_{5,6}\,.$$



Graphs obtained from these expressions are isomorphic, which is easy to make sure directly. From expression a) it results that a given topology is functioning in reference to removal of any two nodes and any two edges. Additionally, as follows from this notion from the first argument (the independent variable of function) we can remove four edges, from the second argument we can remove one edge and vice versa.

Really unambiguous connection gives coherent a graph, even if one of the arguments is a coherent graph. Removal of all edges in the first (second) argument means replacement of graphs $T_{i,j}$ by other graphs T_i^0 and T_j^0 properly (i,j = 1,2,3,4) and then the graph $\left(T_1^0 \cup T_2^0 \cup T_3^0 \cup T_4^0\right) * f\left(T_{5,6} \cup T_{7,8} \cup T_{8,5}\right)$ will be coherent.

Note that from the expression a) it results that in a given graph it is possible to remove all nodes belonging to one wall and the outcome graph will be coherent. Indeed in this case, expression a) receives the form $T_{1,2} * fT_{5,6}$ and if both arguments are coherent graphs then the result will be a coherent graph.

Expression b) for this graph shows how the hypercube arises.

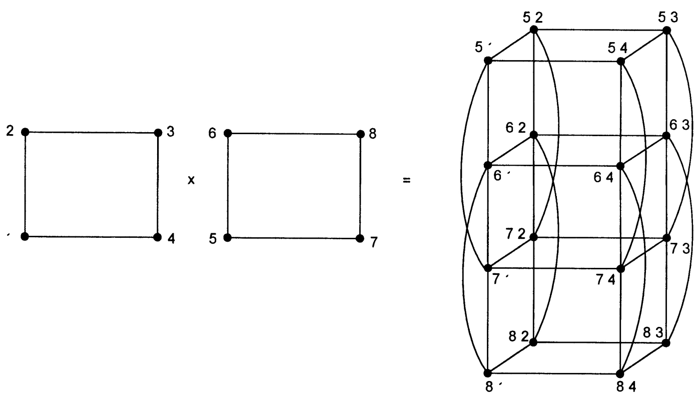
4) *Hypercube.* The graph lying on the basis of this topology is a hypercube (Fig. 4).

It is possible to receive more thorough detailed characteristics if an algebraic expression is analysed for this graph. The algebraic expression is as follows:

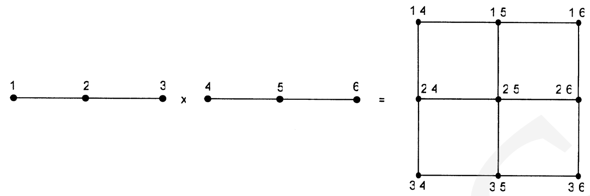
$$\left(T_{1,2} \cup T_{2,3} \cup T_{3,4} \cup T_{4,1}\right) \times \left(T_{5,6} \cup T_{6,7} \cup T_{7,8} \cup T_{8,5}\right)$$

As follows from this expression removing two arbitrary adjacent nodes in the arbitrary argument lead to the shared graph which is isomorphic to the cube. And from this we obtain, that removing arbitrary edges (nodes of these edges and two arbitrary edges) does not infringe the consistency of the hypercube [5,6].

Besides, removing an arbitrary edge in the arbitrary argument in the expression quoted above means the coherent graph remains. It means that it is possible to remove 8 edges in the final graph which correspond to the removed edge. The maximum number of edges which we can be removed from this graph equals $d = |E| - |V| + 1 = 32 - 16 + 1 = 21$. From this estimation follows, that this graph is more reliable in the relation to operation of removing the edge than cube.



5) *Mesh. N* – the dimensional mesh on the basis of this topology (Fig. 5, two-dimensional-ness of the mesh was taken for simplicity).



The algebraic expression for this graph is as follows:

$$\left[\left(T_{1,2} \cup T_{2,3}\right) *_{f1} \left(T_{4,5} \cup T_{5,6}\right)\right] \cup \left[\left(T_{7,8} \cup T_{8,9}\right) *_{f2} \left(T_{4,5} \cup T_{5,6}\right)\right],$$

where $f_1(i) = i + 3$, $f_2(j) = j + 3$, $i = 1,2,3$, $j = 4,5,6$.

From this expression, it can be seen that removing the arbitrary node in one of the arguments and the suitable node in the other argument means that the coherent graph remains. That means that it is possible in the final graph to delete two arbitrary adjacent nodes without infringing coherency of the graph. It is possible to present the given graph in the different way:

$$\left(T_{1,2} \cup T_{2,3}\right) x \left(T_{4,5} \cup T_{5,6}\right).$$

As follows from this expression it is not allowed to remove a single edge since the accident graph will be incoherent. It means that arguments of this product can not be reduced to simpler coherent graphs which form those grids. It is possible, in the analogous way to analyse topology, at the base of which de Bruijn's and Kautz's graphs be fulfilling certain conditions.

Conclusions

Algebra presented in this work was applied for research of the property of various topologies in the process of designing the computer network. The basic advantage of the presentation of network topology in the form of algebraic expression is in the fact that from this presentation there can be seen what elements should participate in the construction of the network and what is structure of these elements and reliability (resistance) of networks compared to disadvantages when particular components will fall out of the move.

References

[1] Hajder M., Dymora P., *Algorithmical and topological methods of fault tolerance assurance*, Annales UMCS Informatica, (2004).
[2] Krivoi S., Hajder M., Dymora P., Mazurek M., *The matrix method of determining the fault tolerance degree of a computer network topology*, KDS-2005, Boolgaria, Varna 2005,
[4] Ахо А., Хопкрофт Дж., Ульман Дж. *Построение и анализ вычислительных алгоритмов*, М: Мир., (1979) 535, in Russian.
[3] Свами М., Тхуласираман К., *Графы, сети и алгоритмы*. М: Мир., (1984) 454, in Russian.
[5] Hajder M., Loutskij G., Streciwilk W., Informatyka. Rzeszow: WSIZ., (2002) 714, in Polish.
[6] Hajder M., Mazurek M., Dymora P., Designing of multichannel optical communication system topologies criteria optimization, Annales UMCS Informatica, (2003).