UDC 004.021+004.722

Rozhkov A.N.
Galishnikova V.V.

Digital Platform for Life Support Management: Topological Concept

In modern digital control platforms, reliability is achieved through the ability to reroute flows in the event of breakdowns and failures. To achieve this, it is possible to use a topological approach that considers the interrelations of elements in various systems, including life support. An overview of existing flow routing algorithms is given; their advantages and disadvantages are given. The paper presents a concept of flow management based on topological tables. Formal concepts of rank (node – edge – face – cell hierarchy) are introduced. For each pair of ranks, a topological contact relation is introduced, and the resulting matrix of sixteen topological tables T_{km} records the incidence of elements without using metric information. Such a structure forms a compact, stable connectivity storage, simplifying network updating and analysis. Then, we consider how, based on these concepts, it is possible to organize the storage of connectivity information (topological tables) and use them to reroute data flows in the event of node or channel failures, increasing network reliability. A description of the flow rerouting algorithm is given. After a node or link failure is detected, the corresponding records in the tables are marked as inactive, then the affected sessions are automatically identified, and the shortest bypass route is selected for each. The sequence of operations includes:

1. Prompt table updating.
2. Classification of the affected flows.
3. Search for an alternative path or selection of a previously saved backup.
4. Instant switching to an alternative route without inter-node exchange of service messages.

If backup routes are available, the downtime is limited by the event detection delay; otherwise, the path is calculated using the updated graph with linear complexity.

Keywords: *digital control platforms, redirection of information flows, maintaining operability during failures, topological tables, topological contact, network reliability, neighbourhood*

Rozhkov Aleksandr Nikolayevich,
Senior Lecturer at the Department of Information Science and Applied Mathematics; Moscow State University of Civil Engineering (National Research University) (MGSU); 26 Yaroslavskoe shosse, Moscow, 129337, Russian Federation; SPIN-code: 4052-7467, AuthorID: 1146492, Scopus AuthorID: 58038282400, WoS ResearcherID: AFR-7324-2022, ORCID: 0000-0002-0729-5644; rozhkovalex@hotmail.com

Galishnikova Vera Vladimirovna,
Doctor of Technical Sciences, Associate professor, Professor; Moscow State University of Civil Engineering (National Research University) (MGSU); 26 Yaroslavskoe shosse, Moscow, 129337, Russian Federation; Professor; Russian University of Friendship of Peoples named after Patrice Lumumba (RUDN); 6 Miklukho-Maclayst., Moscow, 117198, Russian Federation; SPIN-code: 2765-7069, AuthorID: 294267, Scopus AuthorID: 55303553900, WoS ResearcherID: AAA-7515-2022, ORCID: 0000-0003-2493-7255; galishni@yandex.ru

INTRODUCTION

The importance of detecting faults in low-current, fire-fighting, etc. systems

In today's world, the safety and reliability of engineering systems play a key role in both the private and industrial sectors. Low-current systems, fire-fighting systems, and communication systems represent complex technological solutions that ensure the uninterrupted operation of critical facilities, the safety of people, and the preservation of property. However, even a minor malfunction in these systems can lead to serious consequences, including disruptions in production processes, material losses, and — most critically — a threat to life and health.

Detecting faults at early stages not only minimizes risks but also significantly reduces repair and maintenance costs. For the effective functioning of all these systems, it is necessary to implement methods for diagnosis, monitoring, and prevention of potential problems. Modern technologies such as automated control systems, intelligent data analysis algorithms, and specialized equipment help to increase the accuracy and timeliness of fault detection.

One of the promising directions for improving diagnostic systems is the development of a new method for effective resource redistribution based on a topological approach. This method allows one to take into account the structure and interrelations of system elements, which contributes to a more rational distribution of loads, a reduction in the risk of failures, and an increase in overall infrastructure reliability. The application of the topological approach enables the identification of potential bottlenecks, the prediction of possible faults, and the optimization of resource usage to enhance system efficiency.

This article examines the importance of timely fault detection in low-current systems, fire-fighting systems, and communication systems, and proposes an innovative approach to resource redistribution based on topology. Special attention is paid to the main causes of problems, methods for their elimination, and practical techniques that help to improve the reliability and efficiency of these systems.

OVERVIEW OF EXISTING ALGORITHMS

Currently, there are many algorithms aimed at the effective redistribution of resources in low-current systems, fire-fighting systems, and communication systems in emergency situations. Among the most common methods are:

Dynamic Resource Redistribution Algorithms: these methods are based on real-time monitoring of system status and decision-making based on changing parameters. They are widely used in automatic control systems, allowing for rapid response to emergencies [1, 2]. A disadvantage of this approach is the high computational load and the need for constant data collection and processing, which may lead to delays unacceptable in emergency situations.

Graph Routing Algorithms: used in network and communication systems for finding optimal paths for data transmission and load redistribution [3, 4]. Examples include Dijkstra's algorithm, Bellman-Ford [5], and their modifications [6–8]. Their disadvantages, similar to those of dynamic resource redistribution, are the high computational complexity when scaling the network and limited adaptability to sudden failures.

Optimization Methods Based on Queueing Theory: these methods regulate data flows and allocate the bandwidth of communication systems [9, 10]. They minimize delays and data losses in emergency situations. However, they are dependent on input parameters and are complex to adapt to dynamically changing conditions.

Artificial Intelligence and Machine Learning Methods: algorithms using neural networks and deep learning are capable of predicting potential faults and redistributing resources in advance to minimize damage. Their explosive increase in efficiency is associated with the development of electronics and parallel computing [11–14]. However, these methods require large amounts of training data, produce decisions that can be difficult to interpret, and involve lengthy training processes.

Hybrid Algorithms: these combine several approaches, allowing for the consideration of various risk factors and adaptation to specific conditions [15–17]. Such algorithms are used in complex engineering systems where a high degree of autonomy and precision is required. However, the combination of different approaches also brings together the disadvantages of each method, such as complexity of implementation, high computational load, and the need for fine tuning.

The application of efficient resource redistribution algorithms plays an important role in ensuring the uninterrupted operation of low-current systems, fire-fighting systems, and communication systems. However, most existing approaches have certain drawbacks related to high computational complexity, low adaptability, and dependence on external parameters. Using a topological approach helps to eliminate these problems, increase system predictability, optimize data routing, and improve overall resource distribution, ultimately enhancing network reliability [18]. The main outcome of its use is the ability to predict and preempt bottlenecks, which reduces the load on the computational system and speeds up the system's response. While each group of methods has its individual advantages, the topological approach can be used with graph algorithms for predicting and reserving critical nodes; in the analysis of optimization methods based on queueing theory, the proposed approach allows for dynamic load transfer and identification of backup routes; and for artificial intelligence and machine learning methods, topological information becomes available as additional features for more accurate prediction. Simplification of the structural organization of algorithms through the use of topological data for more logical routing and resource distribution becomes possible when employing a topological approach for hybrid algorithms.

Despite the fact that the application of the topological approach can eliminate most of the shortcomings of the considered methods, the aim of this work is to develop a new, "inherently" topological algorithm.

MATERIALS AND METHODS

Statement of the problem

The goal of this work is to develop a new algorithm based on topological tables [19, 20] that is free from the disadvantages of the aforementioned methods. In the proposed method, the authors aim to move away from describing topology by traditional programming methods — which represent the relationships between elements and objects using real numbers — and towards a purely topological method. This is motivated by the fact that processing and interpreting real numbers on a computer always lead to inaccuracies.

Hierarchy of Components

To formalize the network, we introduce a hierarchy of topological elements by rank (dimension):

- **Node (Vertex):** a zero-rank (0-dimensional) element. It denotes a basic object (for example, a device, a node computer, or a switch). The set of all nodes is denoted by *N*;
- **Edge (Link):** a first-rank (1-dimensional) element. It represents the connection between two nodes through which flows can pass (for example, a cable or a communication channel). Formally, an edge e connects a pair of nodes (n_1, n_2), with (n_1, n_2) $\in N$. The collection of all edges is denoted by *E*;
- **Face (Surface):** a second-rank (2-dimensional) element. In the context of networks, this may correspond to a closed cycle of links forming an alternative route. A face is defined as a closed sequence of edges that form a loop. For example, if the nodes and links of a network form a ring, the faces represent such ring topologies. The set of faces is denoted by *F*;
- **Cell (Volume):** a third-rank (3-dimensional) element. This usually corresponds to a closed volume bounded by faces. In network models, cells occur less frequently but may describe, for instance, autonomous subnets or isolated segments completely surrounded by links. The set of cells is denoted by *C*.

Definition 1: The *rank* of a topological element is the dimension of the corresponding object: nodes have rank 0, edges rank 1, faces rank 2, and cells rank 3. This hierarchical model corresponds to representing the network as a cellular complex, where nodes are 0-dimensional vertices, edges are 1-dimensional arcs, faces are 2-dimensional polygons (cycles), and cells are 3-dimensional regions. This hierarchy allows for a formal description of the network structure and the interrelations between its elements at different levels.

Geometric Contact

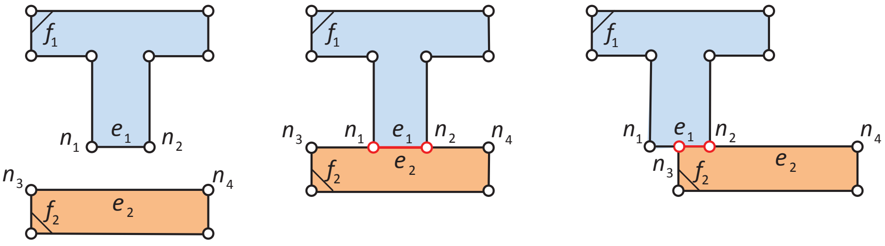
Geometric contact: two elements, whose boundaries have common points, but which do not have any other common point, are in geometric contact. Fig. 1 shows examples of geometric contact of a *T*-shaped face f_1 and a rectangular face f_2.

In the centre diagram of Fig. 1, nodes n_1, n_2 and edge e_1 are subdomains(subset) of face f_1 but not of face f_2. The points of nodes n_1, n_2 and edge e_1 are interior points of edge e_2 of face f_2. Faces f_1 and f_2 are in geometric contact, but not in topological contact. The boundaries of the two domains do not contain a common domain.

In the right diagram of Fig. 1, node n_3 and the red line segment are not domains of the boundary of face f_1. Node n_3 and the redline segment are not domains of the boundary of face f_2. The boundaries of the two faces do not have a common domain. Faces f_1 and f_2 are in geometric contact, because their boundaries have common points, but the faces do not touch topologically.

Topological Contact and Adjacency of Elements

A key concept in the structural description is topological contact — the incidence relationship between elements of different ranks. An element of a lower rank is said to be in topological



No contact Geometric contact Geometric contact

Fig. 1. Geometric contact between faces f_1 and f_2

contact with an element of a higher rank if it is part of its composition (i.e., it forms part of its boundary):

- a node is in topological contact with an edge if the node is one of the endpoints (vertices) of that edge. In other words, node n is incident to edge *e*, if $n \in e$;
- an edge is in topological contact with a face if the edge is part of the cycle forming that face. Formally, edge *e* is incident to face *f*, if $e \in f$ (the edge lies on the boundary of the face);
- a face is in contact with a cell if the face is part of the boundary of that cell (the face forms part of the surface enclosing the cell).

Thus, topological contact defines a "part – whole" relation: a node is included in an edge, an edge is part of a face, and a face bounds a cell. This relation is bidirectional: for example, if node *n* is in contact with edge *e* (node *n* — endpoint of *e*), then edge *e* is also in contact with node *n*.

In addition to the incidence of different ranks, one can introduce the notion of adjacency among elements of the same rank based on contact: two elements of the same type are considered adjacent if they share a common element of a lower rank. For instance, two edges are considered adjacent if they share a common node; two faces are adjacent if they share at least one edge; and so on. For example, if edges $e_1 = (n_1, n_2)$ and $e_2 = (n_2, n_3)$ share node n_2, then they are topologically adjacent via node n_2.

Definition 2: topological contact is the relation $\mathbf{\Gamma}$, $\mathbf{\Gamma}(\mathbf{X}, \mathbf{Y})$ between elements of different ranks, for which $\mathbf{\Gamma}$, $\mathbf{\Gamma}(\mathbf{X}, \mathbf{Y})$ is true if element $\mathbf{X}$ is included in (or forms part of the boundary of) element $\mathbf{Y}$. In practice, the most important contacts are those between nodes and edges (node – edge) and between edges and faces (edge – face), as they describe network connections and cycles, respectively.

The presence of an extensive structure of contacts means that the network possesses alternative connections: for example, if two nodes are connected by several different edges through various intermediate nodes (i.e., there exists a cycle/face linking them), then if one of the nodes or edges fails, the network can remain connected through other elements. It is precisely the use of topological contacts (incidences) that allows one to identify such backup routes.

Topological Tables for Flow Management

For effective management of data flows in a network and their redirection in case of failures, it is necessary to have a data structure that stores complete information about the network's topology — who is connected to whom and via which cycles. A topological table is a collection of data about the topological contacts throughout the network, organized in a convenient form (for example, as adjacency lists or incidence matrices).

Essentially, a topological table contains a map of the network: for each node, all incident edges (and the neighboring nodes connected via those edges) are specified; for each edge, its terminal nodes and possibly the cycles it belongs to are recorded; and so on. In its simplest form, a topological table can be implemented as a dictionary or adjacency table: each node is mapped to a list of neighboring nodes (directly connected via an edge). Additional information about cycles may also be stored — for instance, identifiers of the cycles (faces) in which each edge participates, which allows for the detection of alternative bypass routes.

Example of topological table's structure: consider a network with nodes $N = \{n_1, n_2, n_3, n_4\}$ and edges $E = \{\{n_1, n_2\}, \{n_2, n_3\}, \{n_3, n_4\}, \{n_4, n_1\}\}$, forming a cycle $f_1 = n_1 - n_2 - n_3 - n_4 - n_1$ (face f_1).

A topological adjacency table

Node	Neighboring Nodes (via an Edge)
n_1	n_2, n_4
n_2	n_1, n_3
n_3	n_2, n_4
n_4	n_1, n_3

Here, for example, the row for node n_1 shows that n_1 is in topological contact with nodes n_2 and n_4 (via edges $\{n_1, n_2\}$, and $\{n_4, n_1\}$, respectively). Similarly, the table lists the direct connections for each node. Such a table is essentially equivalent to storing the node – edge incidence lists, but in a human-readable form that displays the entire connectivity structure.

Another way to represent a topological table is to list all known routes in the network, indicating both the primary and backup paths. For any destination node, one can store not only the optimal route but also one or several alternate routes. This concept is used in some routing protocols. For instance, the EIGRP protocol builds a topological table at each router, in which both the best (primary) route to each network destination and alternative routes are stored [21]. Under normal operation, traffic follows the primary route, but if the primary path becomes unavailable, the protocol immediately switches to the backup route from the topological table without additional computations or message exchanges with neighbors. This mechanism significantly reduces the time required to restore network connectivity in the event of failures.

Definition 3: a topological table is a data structure containing complete information about the network topology (the topological contacts between nodes, edges, faces, etc.) used for determining the routes for data flows. Formally, a topological table can be represented as a set of mapping: $\tau_0 : N \to 2^E$ (one that maps each node to the set of incident edges), $\tau_1 : E \to 2^N$ (maps each edge to the pair of terminal nodes), $\tau_2 : E \to 2^F$ (maps an edge to the set of faces it

T_{km}		Rank *m*, component type			
		1 node	2 edge	3 face	4 cell
Rank *k*, component type	1 node	T_{11}	T_{12}	T_{13}	T_{14}
	2 edge	T_{21}	T_{22}	T_{23}	T_{24}
	3 face	T_{31}	T_{32}	T_{33}	T_{34}
	4 cell	T_{41}	T_{42}	T_{43}	T_{44}

Face – edge – table						
Face	Edges of the face					
1	1	2	3	4	–	–
2	1	31	7	25	–	–
5	4	49	23	24	5	26
6	7	9	10	8	6	5
7	6	25	22	24	–	–
12	7	32	58	27	–	–
13	58	35	55	30	–	–

Fig. 2. Matrix of the topological tables for linear complexes

belongs to), and so on. Based on this data, one can derive the list of neighbors for each node and identify the presence of alternative paths (cycles) between nodes.

Thus, the topological table serves as the basis for flow management algorithms. With such a table, the system can solve the routing problem: choosing optimal paths for flows based on the current network state and switching flows to alternative paths when the state changes (for example, upon failures). Unlike local routing tables (which store only the next hop for a destination), the topological table contains a global view of the network's connectivity, enabling flexible route reconfiguration and the discovery of bypass solutions. The following sections describe algorithms for redirecting flows in case of failures based on topological tables.

The matrix in Fig. 2 shows the 16 types of topological tables that can be constructed for a linear complex with node, edge, face and cell elements. The symbol T_{km} denotes a table, whose pivot elements have rank k and whose contact elements have rank m. For example, table T_{32} in Fig. 2 is the topological face–edge–table. Table T_{23} is the topological edge–face–table. The diagonal elements of the matrix in Fig. 2 are tables whose pivot and contact element are of equal rank.

RESULTS

Algorithm for Flow Redirection in the Event of a Node Failure

A node failure is a situation in which a particular network node suddenly stops functioning (for example, due to a device power-off or a node software fault). Formally, when node $X \in N$ ceases to function, all edges $e \in E$ incident to X (i.e. for which $X \in e$) also cease to operate. As a result, the removal of X disrupts all flows that passed through it. The task of flow management is to promptly restore these flows by redirecting them along alternative routes, if available in the topology.

The redirection algorithm can be divided into several stages:

1. **Updating the Topological Table:** immediately after detecting the failure of node *X*, the system updates the topological table by marking node *X* as non-operational and excluding all edges incident to *X*. This leads to changes in the neighbor lists: all nodes that were previously neighbors to *X* lose their connection with it. Formally, this corresponds to removing vertex *X* and all its incident edges from the network graph $G = (N, E)$, resulting in a new graph $G' = (N \backslash X, E \backslash e: X \in e)$.

2. **Identifying Affected Flows:** the flows whose routes included the failed node *X* are identified. Typically, these are flows where *X* served as an intermediate node (a router) rather than being the sender or receiver. Based on the topological table, one can find pairs of nodes whose connection passed through *X*. If the topological table already contains both primary and alternative routes for each destination, then at this stage it is simply determined which primary routes passed through *X*, and the corresponding backup routes are retrieved from the table.

Flow Redirection: for each affected flow, a new route that bypasses node *X* is calculated. There are two approaches:

- *Precomputed Routes*: if the topological table already stored alternative paths (backup routes) for the given directions, the system immediately switches to them. In this case, the restoration delay is minimal — it is sufficient to update the local routing tables on the neighbors of *X*, replacing *X* with the next node from the backup route;
- *Dynamic Route Recalculation*: if alternatives were not precomputed, the system initiates a path-finding algorithm on the updated graph G'. Typically, a shortest path algorithm (for example, Dijkstra's algorithm or breadth-first search) is used based on the updated topological table. Since the table contains complete connectivity information, the source node or a designated network controller can establish a new route from the sender to the receiver that bypasses the failed node *X*. The newly found alternative path is then distributed to the corresponding nodes as the new route for that flow.

4. **Restoration of Transmission:** the updated routes take effect, and data is redirected along the new paths. The flow is restored — packets reach their destination by bypassing the segment affected by the failure.

It is important to note that the feasibility of step 3 (the existence of an alternative path) directly depends on the original topology. If the failed node was an articulation point of the network (i.e., its removal splits the graph into several disconnected components), then no alternative route exists within G'. Formally, node *X* is an articulation point if, upon its removal, the graph G' breaks into two or more connected components. In such a case, some nodes become completely isolated, and the flows to them cannot be restored (until the physical problem is resolved). A topology is considered robust if it lacks such articulation points (i.e., the graph is biconnected by vertices) — in that case, the network withstands the failure of any single node without losing connectivity. Similarly, the absence of bridges (edges whose removal would disconnect the graph) signifies resilience to individual link failures. The design of a reliable communication system must ensure the required degree of connectivity (for example, biconnectivity of the network graph), and flow management must efficiently utilize this connectivity for routing.

With the presence of backup connections, the topological approach ensures high reliability: flows automatically "bypass" the failed node via other routes embedded in the network structure.

The recovery time is limited either to the immediate switch to a precomputed backup route or the time needed to compute a new path. Because the topological table stores global network information, the search for alternatives is performed locally (within a controller or router) without the need to query remote nodes for their status — thereby accelerating the system's reaction to a failure. For example, in self-healing networks based on graphs, route state converges faster if each node is already aware of the existence of bypass paths [21]. Reducing the downtime of flow increases the overall fault tolerance of the communication system.

Example: Restoring a Flow After a Node Failure

Consider a specific example illustrating the application of a topological table for redirecting a flow in the event of a node failure. Suppose there is a network of four nodes (*A*, *B*, *C*, *D*) connected in a ring (each node is connected to two neighbors): *A–B*, *B–C*, *C–D*, *D–A*. Initially, traffic between nodes follows the shortest paths. Assume that node *B* is designated as the intermediate router for the data flow from *A* to *C* (i.e., data from *A* is routed through *B* to *C*). Under normal conditions, node *D* remains idle for this flow, even though it is physically part of the ring.

Now, suppose node *B* fails (for example, due to a sudden power outage or software error). Node *B* and both its incident channels (*A–B* and *B–C*) cease to function. From a topological standpoint, *B* becomes unavailable, and the ring opens up into a chain *A–D–C*. Nodes *A* and *C* lose their direct connection with *B* but remain connected to each other via node *D*. The data flow from *A* to *C*, interrupted at the moment of *B*'s failure, can be restored through the alternative path.

In the presented example (Fig. 3, 4), the failure affects only one flow (between *A* and *C*). After rerouting through *D*, data continues to be delivered. The flow's downtime is determined solely by the time required to detect the failure of *B* and select an alternative route. If the network did not have the redundant link to *D* (for instance, if the topology were simply a chain *A–B–C* without node *D*), then the failure of *B* would make communication between *A* and *C* impossible — illustrating the importance of planning a topology without articulation points.

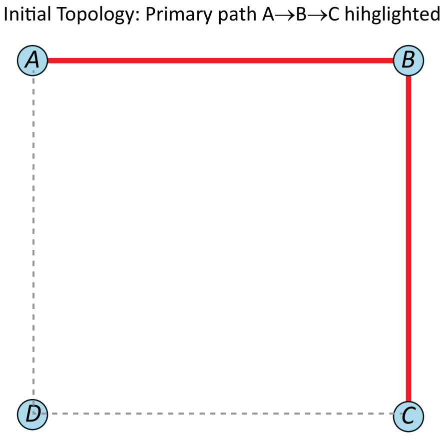


Fig. 3. The primary network topology of the four-node ring (A, B, C, D) with the main flow from A to C (highlighted in red via node B). The grey dashed lines indicate the network's backup links (in this case, the alternative A–D–C path) which are not used during normal operation. At this stage, the topological table contains the primary route A→B→C for the A→C flow along with information about the existence of an alternative cycle A–D–C

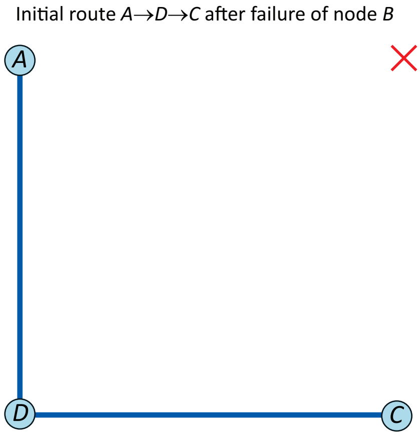


Fig. 4. The network after the failure of node B. Node B (located in the upper right) is marked with a red cross to indicate its failure. The links A–B and B–C are broken. However, thanks to the ring topology, nodes A and C remain connected through D (the route A→D→C is highlighted in blue). Immediately after the failure, the topological table is updated: node B is removed fromthe neighbor lists. The new neighbor lists become: A: {D}, C: {D}, D: {A, C}. Based on these data, the system detects that a path between A and C still exists (via D). The route A→D→C is either retrieved from the topological table (if it was precomputed as a backup) or computed anew, and the flow from A to C is redirected through node D, bypassing the failed B

DISCUSSION

The considered ring topology exhibits redundancy — each node has a degree of 2 (two links), and the graph is biconnected. The removal of any single node still leaves the graph connected. Therefore, the probability of maintaining connectivity between any pair of nodes is higher than in a linear topology. In such a network, topological tables store alternative routes (for example, a direct route through one neighbor and a bypass route through the opposite node), which enhances the network's self-healing capability. In real networks, connections can also be interrupted for other reasons (link failure, node overload, software error), but the principle remains the same: the more alternative paths that are provided for and reflected in the topological tables, the higher the network's resilience. Flow management based on topological tables allows for the automatic selection of these routes, maintaining service even in the face of partial infrastructure failures.

CONCLUSION

Flow management methods based on topological tables demonstrate effectiveness under conditions of node and channel failures. The formal representation of the network through the hierarchy "node – edge – face – cell" and the concept of topological contact provide a rigorous foundation for routing algorithms. Topological tables serve as the central element linking theory and practice: they store the structural information about the network necessary for rapid flow redirection. The inclusion of backup routes in topological tables enables the system to react instantly to failures, switching to alternative paths without lengthy route recalculation. Even if alternatives are not precomputed, the availability of the complete topology accelerates the search for a new route. Thus, the topological approach to flow management enhances the reliability of the communication system — the failure of individual components does not lead to catastrophic network breakdowns, as issues are localized and bypassed. This is especially critical for low-current systems and automated diagnostic platforms where continuous and timely data transmission is essential.

REFERENCES

1. Zhao Z., Zhang P., Yang H. Cascading failures in interconnected networks with dynamical redistribution of loads. Physica A: Statistical Mechanics and its Applications. 2015; 433:204-210. DOI: 10.1016/j.physa.2015.03.030.

2. Gao C., Fan Y., Jiang S., Deng Y., Liu J., Li X. Dynamic robustness analysis of a two-layer rail transit network model. IEEE Transactions on Intelligent Transportation Systems. 2021; 23(7):6509-6524. DOI: 10.1109/TITS.2021.3058185

3. Bejerano Y., Breitbart Y., Orda A., Rastogi R., Sprintson A. Algorithms for computing QoS paths with restoration. IEEE/ACM Transactions on Networking. 2005; 13(3):648-661. DOI: 10.1109/TNET.2005.850217

4. Wagner D., Willhalm T. Speed-up techniques for shortest-path computations. In Annual Symposium on Theoretical Aspects of Computer Science. 2007; 23-36. Berlin, Heidelberg, Springer Berlin Heidelberg. DOI: 10.1007/978-3-540-70918-3_3

5. Dinitz Y., Itzhak R. Hybrid bellman–ford–dijkstra algorithm. Journal of Discrete Algorithms. 2017; 42:35-44. DOI: 10.1016/j.jda.2017.01.001

6. Jepsen M.K., Petersen B., Spoorendonk S., Pisinger D. A branch-and-cut algorithm for the capacitated profitable tour problem. Discrete Optimization. 2014; 14:78-96. DOI: 10.1016/j.disopt.2014.08.001

7. Nejad M.M., Mashayekhy L., Chinnam R.B., Phillips A. Hierarchical time-dependent shortest path algorithms for vehicle routing under ITS. Iie Transactions. 2016; 48(2):158-169. DOI: 10.1080/0740817X.2015.1078523

8. Pathan A.S.K., Monowar M.M., Khan S. Simulation Technologies in Networking and Communications: Selecting the Best Tool for the Test. CRC Press, 2014.

9. Alam M.G.R., Suma T.M., Uddin S.M., Siam M.B.A.K., Mahbub M.S., Hassan M.M. et al. Queueing theory based vehicular traffic management system through Jackson network model and optimization. IEEE Access. 2021; 9:136018-136031. DOI: 10.1109/ACCESS.2021.3116503

10. Afolalu S.A., Ikumapayi O.M., Abdulkareem A., Emetere M.E., Adejumo O. A short review on queuing theory as a deterministic tool in sustainable telecommunication system. Materials Today, Proceedings. 2021; 44:2884-2888. DOI: 10.1016/j.matpr.2021.01.092

11. Petrović A., Nikolić M., Bugarić U., Delibašić B., Lio P. Controlling highway toll stations using deep learning, queuing theory, and differential evolution. Engineering Applications of Artificial Intelligence. 2023; 119:105683. DOI: 10.1016/j.engappai.2022.105683

12. Dai B., Cao Y., Wu Z., Dai Z., Yao R., Xu Y. Routing optimization meets Machine Intelligence: A perspective for the future network. Neurocomputing. 2021; 459:44-58. DOI: 10.1016/j.neucom.2021.06.093

13. Amin R., Rojas E., Aqdus A., Ramzan S., Casillas-Perez D., Arco J.M. A survey on machine learning techniques for routing optimization in SDN. IEEE Access. 2021; 9:104582-104611. DOI: 10.1109/ACCESS.2021.3099092

14. Sivalingam K.M. Applications of artificial intelligence, machine learning and related techniques for computer networking systems. arXiv preprint arXiv: 2105.15103. 2021. DOI: 10.48550/arXiv.2105.15103

15. Liang G., Kintak U., Ning X., Tiwari P., Nowaczyk S., Kumar N. Semantics-aware dynamic graph convolutional network for traffic flow forecasting. IEEE Transactions on Vehicular Technology. 2023; 72(6):7796-7809. DOI: 10.1109/TVT.2023.3239054

16. Duan Y., Chen N., Shen S., Zhang P., Qu Y., Yu S. FDSA-STG: Fully dynamic self-attention spatial-temporal graph networks for intelligent traffic flow prediction. IEEE Transactions on Vehicular Technology. 2022; 71(9):9250-9260. DOI: 10.1109/TVT.2022.3178094

17. Srilakshmi U., Veeraiah N., Alotaibi Y., Alghamdi S.A., Khalaf O.I., Subbayamma B.V. An improved hybrid secure multipath routing protocol for MANET. IEEE Access. 2021; 9:163043-163053. DOI: 10.1109/ACCESS.2021.3133882

18. Rozhkov A., Galishnikova V. Explicit Digital Models of Linear Complexes. International Journal for Computational Civil and Structural Engineering. 2022; 18(4):101-110. DOI: 10.22337/2587-9618-2022-18-4-101-110. EDN RZOCBU

19. Rozhkov A.N., Galishnikova V.V. Construction and Modification of Topological Tables for Digital Models of Linear Complexes. Mathematical and Computational Applications. 2023; 28(2):37. DOI: 10.3390/mca28020037

20. Rozhkov A.N., Galishnikova V.V. Construction of Topological Tables for Digital Models of Linear Complexes. BIM IN CONSTRUCTION & ARCHITECTURE. 2023; 113-119. DOI: 10.23968/BIMAC.2023.016

Цифровая платформа управления жизнеобеспечением: топологическая концепция

В современных цифровых системах управления надежность обеспечивается способностью перенаправлять потоки при сбоях и отказах. Для этого можно применять топологический подход, учитывающий взаимосвязи элементов в различных системах, включая системы жизнеобеспечения. Представлен обзор существующих алгоритмов маршрутизации потоков, отмечены их достоинства и недостатки.

В работе предлагается концепция управления потоками на основе топологических таблиц. Вводятся формальные понятия рангов (иерархия «узел – ребро – грань – ячейка»). Для каждой пары рангов задается топологическое отношение контакта, и результирующая матрица из шестнадцати топологических таблиц Tkm фиксирует инцидентность элементов без использования метрической информации. Такая структура образует компактное устойчивое хранилище связности, упрощая обновление и анализ сети.

Далее рассматривается, как с помощью этих понятий можно организовать хранение информации о связности и применять ее для перенаправления потоков данных при отказах узлов или каналов, повышая надежность сети. Описан алгоритм перенаправления: после обнаружения отказа соответствующие записи в таблицах помечаются как неактивные, затем автоматически идентифицируются затронутые сеансы и для каждого выбирается кратчайший обходной маршрут. Последовательность операций включает:

1. Оперативное обновление таблиц.
2. Классификацию затронутых потоков.
3. Поиск альтернативного пути или выбор ранее сохраненного резервного.
4. Мгновенное переключение на альтернативный маршрут без межузлового обмена служебными сообщениями.

При наличии резервных маршрутов длительность простоя ограничивается задержкой обнаружения события; при их отсутствии путь вычисляется по обновленному графу с линейной сложностью.

Ключевые слова: *цифровые платформы управления, перенаправление информационных потоков, поддержание работоспособности при отказах, топологические таблицы, топологический контакт, надежность сети, соседство*

СПИСОК ИСТОЧНИКОВ

1. Zhao Z., Zhang P., Yang H. Cascading failures in interconnected networks with dynamical redistribution of loads // Physica A: Statistical Mechanics and its Applications. 2015. Vol. 433. Pp. 204–210. DOI: 10.1016/j.physa.2015.03.030
2. Gao C., Fan Y., Jiang S., Deng Y., Liu J., Li X. Dynamic robustness analysis of a two-layer rail transit network model // IEEE Transactions on Intelligent Transportation Systems. 2021. Vol. 23. No. 7. Pp. 6509–6524. DOI: 10.1109/TITS.2021.3058185
3. Bejerano Y., Breitbart Y., Orda A., Rastogi R., Sprintson A. Algorithms for computing QoS paths with restoration // IEEE/ACM Transactions on Networking. 2005. Vol. 13. No. 3. Pp. 648–661. DOI: 10.1109/TNET.2005.850217
4. Wagner D., Willhalm T. Speed-up techniques for shortest-path computations // Annual Symposium on Theoretical Aspects of Computer Science. 2007. Pp. 23–36. DOI: 10.1007/978-3-540-70918-3_3
5. Dinitz Y., Itzhak R. Hybrid bellman–ford–dijkstra algorithm // Journal of Discrete Algorithms. 2017. Vol. 42. Pp. 35–44. DOI: 10.1016/j.jda.2017.01.001
6. Jepsen M.K., Petersen B., Spoorendonk S., Pisinger D. A branch-and-cut algorithm for the capacitated profitable tour problem // Discrete Optimization. 2014. Vol. 14. Pp. 78–96. DOI: 10.1016/j.disopt.2014.08.001
7. Nejad M.M., Mashayekhy L., Chinnam R.B., Phillips A. Hierarchical time-dependent shortest path algorithms for vehicle routing under ITS // Iie Transactions. 2016. Vol. 48. No. 2. Pp. 158–169. DOI: 10.1080/0740817X.2015.1078523
8. Pathan A.S.K., Monowar M.M., Khan S. Simulation Technologies in Networking and Communications: Selecting the Best Tool for the Test. CRC Press, 2014.
9. Alam M.G.R., Suma T.M., Uddin S.M., Siam M.B.A.K., Mahbub M.S.B., Hassan M.M. et al. Queueing theory based vehicular traffic management system through Jackson network model and optimization // IEEE Access. 2021. Vol. 9. Pp. 136018–136031. DOI: 10.1109/ACCESS.2021.3116503
10. Afolalu S.A., Ikumapayi O.M., Abdulkareem A., Emetere M.E., Adejumo O. A short review on queuing theory as a deterministic tool in sustainable telecommunication system // Materials Today: Proceedings. 2021. Vol. 44. Pp. 2884–2888. DOI: 10.1016/j.matpr.2021.01.092
11. Petrović A., Nikolić M., Bugarić U., Delibašić B., Lio P. Controlling highway toll stations using deep learning, queuing theory, and differential evolution // Engineering Applications of Artificial Intelligence. 2023. Vol. 119. P. 105683. DOI: 10.1016/j.engappai.2022.105683
12. Dai B., Cao Y., Wu Z., Dai Z., Yao R., Xu Y. Routing optimization meets Machine Intelligence: A perspective for the future network // Neurocomputing. 2021. Vol. 459. Pp. 44–58. DOI: 10.1016/j.neucom.2021.06.093
13. Amin R., Rojas E., Aqdus A., Ramzan S., Casillas-Perez D., Arco J.M. A survey on machine learning techniques for routing optimization in SDN // IEEE Access. 2021. Vol. 9. Pp. 104582–104611. DOI: 10.1109/ACCESS.2021.3099092
14. Sivalingam K.M. Applications of artificial intelligence, machine learning and related techniques for computer networking systems // arXiv preprint arXiv:2105.15103. 2021. DOI: 10.48550/arXiv.2105.15103
15. Liang G., Kintak U., Ning X., Tiwari P., Nowaczyk S., Kumar N. Semantics-aware dynamic graph convolutional network for traffic flow forecasting // IEEE Transactions on Vehicular Technology. 2023. Vol. 72. No. 6. Pp. 7796–7809. DOI: 10.1109/TVT.2023.3239054
16. Duan Y., Chen N., Shen S., Zhang P., Qu Y., Yu S. FDSA-STG: Fully dynamic self-attention spatial-temporal graph networks for intelligent traffic flow prediction // IEEE Transactions on Vehicular Technology. 2022. Vol. 71. No. 9. Pp. 9250–9260. DOI: 10.1109/TVT.2022.3178094
17. Srilakshmi U., Veeraiah N., Alotaibi Y., Alghamdi S.A., Khalaf A.I., Subbayamma B.V. An improved hybrid secure multipath routing protocol for MANET // IEEE Access. 2021. Vol. 9. Pp. 163043–163053. DOI: 10.1109/ACCESS.2021.3133882
18. Rozhkov A., Galishnikova V. Explicit Digital Models of Linear Complexes, International // Journal for Computational Civil and Structural Engineering. 2022. T. 18. No. 4. Pp. 101–110. DOI: 10.22337/2587-9618-2022-18-4-101-110. EDN RZOCBU
19. Rozhkov A.N., Galishnikova V.V. Construction and Modification of Topological Tables for Digital Models of Linear Complexes // Mathematical and Computational Applications. 2023. Vol. 28. No. 2. P. 37. DOI: 10.3390/mca28020037
20. Рожков А. Н., Галишникова В. В. Построение топологических таблиц для цифровых моделей линейных комплексов //BIM-моделирование в задачах строительства и архитектуры. 2023. С. 113–119. DOI: 10.23968/BIMAC.2023.016
21. Karamela N., Karras D.A. A Comparative Analysis of OSPF and EIGRP Routing Protocol Evaluation // Journal of Transactions in Systems Engineering. 2023. Vol. 1. No. 2. Pp. 73–103. DOI: 10.15157/JTSE.2023.1.2.73-103

Об авторах: **Рожков Александр Николаевич** — старший преподаватель кафедры информатики и прикладной математики; **Национальный исследовательский Московский государственный строительный университет (НИУ МГСУ)**; 129337, г. Москва, Ярославское шоссе, д. 26; SPIN-код: 4052-7467, AuthorID: 1146492, Scopus AuthorID: 58038282400, WoS ResearcherID: AFR-7324-2022, ORCID: 0000-0002-0729-5644; rozhkovalex@hotmail.com;

Галишникова Вера Владимировна — доктор технических наук, доцент, профессор; **Национальный исследовательский Московский государственный строительный университет (НИУ МГСУ)**; 129337, г. Москва, Ярославское шоссе, д. 26; профессор; **Российский университет дружбы народов имени Патриса Лумумбы (РУДН)**; 117198, г. Москва, ул. Миклухо-Маклая, д. 6; SPIN-код: 2765-7069, AuthorID: 294267, Scopus AuthorID: 55303553900, WoS ResearcherID: AAA-7515-2022, ORCID: 0000-0003-2493-7255; galishni@yandex.ru.

For citation: Rozhkov A.N., Galishnikova V.V. Digital Platform for Life Support Management: Topological Concept. *Real Estate: Economics, Management.* 2025; 2:70-76.

Для цитирования: *Рожков А.Н., Галишникова В.В.* Digital Platform for Life Support Management: Topological Concept // Недвижимость: экономика, управление. 2025. № 2. С. 70–76.