An Approach Toward Packet Routing in the OSPF-based Network with a Distrustful Router

KVITOSLAVA OBELOVSKA[1], YAROMYR SNAICHUK[1], JULIUS SELECKY[2], ROSTYSLAV LISKEVYCH[3], TETIANA VALKOVA[1]
[1]Department of Automated Control Systems,
Lviv Polytechnic National University,
Lviv,
UKRAINE

[2]Department of Information Systems,
Comenius University Bratislava,
Bratislava,
SLOVAK REPUBLIC

[3]Ukrainian Industrial Telecommunications LLC,
Lviv,
UKRAINE

Abstract: - Packet routing in computer networks significantly affects the effectiveness of the network, including its security. Various reasons can result in a certain router losing its trust from a security point of view. At the same time, it is necessary to ensure the delivery of packets in the network in general. To address such a situation we propose a new approach to packet routing when a distrustful router appears in the network. The proposed approach completely halts the transmission of transit packets through a distrustful node if it is possible to bypass it. A physical connection to a distrustful node will be used only to transmit packets addressed to this node. If there are no paths to all destinations without using the distrustful node, a path tree is obtained in which the number of nodes that receive packets through the distrustful node is minimized. At the same time, the transmission of packets to all destinations is completely preserved. The outcome of our approach is a path tree that optimizes the routing table, considering the presence of a distrustful router, and minimizes transit flows through this router and the number of physical connections to it.

Key-Words: - minimizing distrustful physical connections; packet routing; shortest paths tree; privacy-preserving routing.

Received: September 5, 2022. Revised: October 7, 2023. Accepted: October 20, 2023. Published: November 3, 2023.

1 Introduction

In computer networks, the process of packet routing is a crucial aspect of ensuring efficient and reliable communication between sender and receiver nodes. Routing is carried out based on routers' routing tables, which, in turn, are built using special routing information exchange protocols.

There is not and cannot be a single universal routing method that would be optimal at the same time for different types and dimensions of networks, different transmission technologies, different types of traffic, and their service requirements. Many works solve routing optimization problems according to specific circumstances and requirements.

In this work, we will consider the problem of a network node losing its security trust. Such nodes will be marked as untrusted and network traffic should omit them to prevent packet reading or Man-in-the-middle attacks. To prevent described issues we propose a new approach to packet routing in computer networks. We aim to ensure the delivery of all packets on the network while mitigating the impact of an unreliable router. When an untrustworthy router is detected, the proposed approach stops transit packets through the router if there are alternative routes to bypass it. Physical connections to an untrusted router will only be used to transmit packets addressed to that particular node. Thus, goals are to optimize the routing table taking into account the presence of an unreliable router, to minimize transit flows through this router and the number of physical connections to it.

The main contributions of this paper relate to the shortest paths tree for router routing tables and can be summarized as follows:

- t
  he method of obtaining a shortest paths tree has been proposed which allows for the minimization of transit flows through a particular node;
- t
  he method for obtaining the shortest paths tree has been developed in which the distrustful node will not be transit if a paths tree with the distrustful node located at the end of a branch;
- f
  or the case when a tree of paths with a distrustful node at the end of a branch does not exist, a method for obtaining the shortest path tree that minimizes the transit flow through the distrustful node has been developed.

2 Related Work

The common approach for construction routing tables is to use a tree of the shortest paths between the sender and destinations, calculated according to a certain criterion or criteria, to ensure fast and reliable service, [1]. One of the directions is that it is necessary to take into account not one, but several, potentially conflicting tasks that must be optimized simultaneously, [2], [3], [4]. To accomplish the goal, different methods, and Shortest Path (SP) algorithms can be used. Routing protocols make use of them to determine the most optimal route for a given packet. Such methods may consider various factors such as network infrastructure, available bandwidth, number of hops, distance, reliability, and others.

In SP algorithms, the pathfinding problem is solved by representing the physical network as a graph, with nodes representing network devices and edges representing the connections between them. To determine the best path for a packet to take, weights are assigned to the edges of the graph, which represent a certain optimization criterion, [5]. In cases where the sender and destination nodes are not directly connected, the pathfinding must take into account the distances between all of the intermediate nodes to determine the most efficient route. This information can be recorded as a label associated with each node in the graph, [6], or stored in a memory table.

We will consider the Dijkstra algorithm as the classic shortest path searching method since it is widely considered a long-established solution for the SP problem. For example, authors in work, [7], employ the Dijkstra algorithm to find the shortest route for tourism purposes. Also, work, [8], presents an example of using the Dijkstra algorithm to find attractive flights for passengers based on different preferences.

OSPF (Open Shortest Path First) protocol is a prominent example of the Dijkstra algorithm usage, [9], in computer networks due to its widespread utilization. This protocol is widely used in various network types, including wired or wireless networks, Autonomous Systems within Wide Area Networks, Wireless Sensor Networks (WSN), and Software Defined Networks (SDN). As shown in work, [10], with modifications to incorporate dynamic network requirements and adapt to changing network paradigms OSPF can be used in SDN, thereby enhancing the flexibility and adaptability of the routing protocol within an SDN environment. Further, OSPF shows better results in throughput and delay than the Routing Information Protocol when tested in SDN, [11]. In addition, OSPF can be used as a basic solution for the routing problem in WSN, [12]. Alternatively, with certain modifications, it enables significant benefits for efficient data transmission, minimal interruptions, and optimal routing for enhanced network performance in WSN, [13].

When OSPF is used with Internet Protocol Security protocol, they provide enhanced security features such as authentication and encryption, ensuring secure and protected communication between network devices, [14]. However, challenges of preventing data transmission through potentially vulnerable or compromised nodes remain and require further attention and improvement. With the rapid advancement of computer networks and millions of devices being connected every year, significant challenges arise in terms of protecting data within the network from malicious hosts, while also providing efficient packet delivery service, [15]. So far, the security of the routing process is an ongoing task.

Different methods for solving network security problems can be suggested. Related works can be divided into two categories based on a technique used to address the presence of untrusted nodes:

- modifying cost calculation function to increase travel costs for untrusted nodes. Thus, minimizing traffic through them. This approach is easy to develop and implement into an existing method that uses the Dijkstra algorithm;
- modifying packet routing algorithm to prevent path calculation for untrusted nodes. This method requires more effort to develop and verify but can prevent any traffic from coming through untrusted nodes.

One of the options is to modify the existing method to include nodes in the network based on their level of trustworthiness, [16], [17]. This approach is based on modifying the cost function for the Dijkstra algorithm to accommodate the trust degree of a node. The trust degree for each node is assigned before path searching and it represents the possibility of malicious use of the transmitting packet by the node. Therefore, the resulting route will include only nodes with a trust degree above a certain configured value for the specific network.

Another similar option is calculating the trust of the node based on the information collected from its neighboring nodes, [18], [19]. In this approach, the trust degree is determined when new nodes are added to the network.

In recent years, the active development of blockchain and machine learning technologies allowed them to find a place in the packet routing process. As suggested in works, [17], [20], [21], [22], blockchain networks can provide load balancing and node security verification based on the history of sending packets inside the network. History is stored in blockchain smart contracts in the shared memory of the network. If a node tries to make a change to the history, it will be considered distrustful by other members of the network. Therefore, this method allows for the detection and bypass of untrusted parts of the network.

An alternative solution can be using Machine Learning (ML) to optimize routing performance and protect the network from security issues, [23], [24]. The growing interest in ML systems caused their usage in developing new routing methods. One such method is described in the work, [23]. It proposes to use a trained model to optimize the routing process with consideration of multiple factors, such as processing time, queuing, transmission, and propagation delays. Authors in works, [25], [26], used the Artificial Intelligence Enabled Routing (AIER) mechanism in SDN to improve the overall network performance by introducing a neural network model to the SDN controller to avoid network congestion. AIER aims to enhance the routing process by intelligently selecting suitable paths to avoid congestion, thereby improving network performance.

In SDN, researchers proposed a method for ensuring secure routing by replicating a minimum specific number of switches or routers from different manufacturers with the highest reliability at the network level. In this method, each switch in the network can be replicated by a structure that contains different switches, with each belonging to a different manufacturer with different reliability, [27]. Authors in work, [28], proposed a Lightweight Routing Authentication mechanism that supports Genetic algorithm-enabled SDN routing in the Internet of Things (IoT) networks, utilizing a blockchain for authentication and route validation. Their system manages diverse IoT clusters, detects malicious nodes, and maintains a Malicious Node's List for route validation purposes.

In WSN, the security of the route can be provided by calculating the trust degree of sensors in the network and further selecting a route with trusted nodes, [29]. In addition, this method can be leveraged to optimize energy consumption in WSN by incorporating energy costs as weights for the network graph, [30].

Highlighted methods are useful when the main concern is network security so they use the rejection of all packets from or to distrustful nodes. The challenge is to ensure all packets are delivered to network nodes without including untrusted nodes. The generated route can be longer but it will be considered safer. Alternatively, network traffic should be sent to the distrustful nodes in case they are the intended recipients or if part of the network can only be accessed through them.

The objective of this work is to develop an innovative method for finding the shortest path tree in a network while considering a compromised node. It aims to improve the security and reliability of network routing by addressing the issues associated with mistrusting a particular network node and minimizing its impact on network security.

3 Materials and Methods

The shortest packet route can be calculated using different routing algorithms. We will consider the problem of single-source routing without negative travel costs. Thus, we can use the classic Dijkstra algorithm as it is already widely used in OSPF-based networks.

In this paper, we present a new privacy-preserving routing method based on the classic Dijkstra algorithm with some significant differences. As an example of a network that will be used to illustrate our method, let's take the weighted directed graph used in works, [6], [30], to show the execution of the Dijkstra algorithm. Since our work is focused on applications in computer networks that usually use full-duplex channels, we replaced the directed graph with an undirected one (Figure 1).

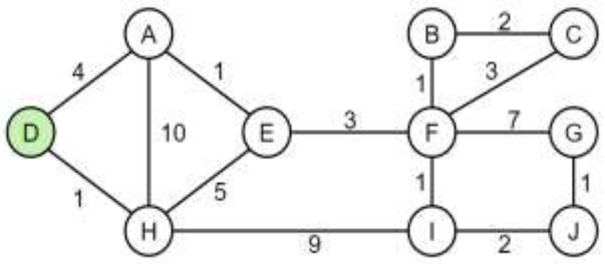

Fig. 1: The example of the network under study, [6], [30].

Graph nodes correspond to routers, edges to full-duplex channels, and numbers near edges to channel metric values. Let node D correspond to the router for which the shortest paths tree must be constructed. The shortest path is the path with the lowest aggregate metric value.

Let's introduce the following notations:

- DS(v) is the sum of the channel weights on the path from root node D to current node v;
- c(w,v) is the sum of the weights of channels between nodes with numbers w and v.

Dijkstra's algorithm for constructing the shortest paths tree, [6], [31], is described below.

The sequence of steps implementing the algorithm can be presented as follows:

1. Form a set of nodes N, which contains the root node in the first step. N = {D}.
2. Set DS(v) = c(D,v), for all nodes v that are neighbors to node D.
3. Find node w that is not in the set N and for which DS(w) is minimum, and add node w to the set N.
4. Update DS(v) for all nodes that are not in the set N. DS(v) = min{DS(v), DS(w)+c(w,v)}.
5. Repeat steps 3-4 until all nodes are in the set N.

In implementing the algorithm, the nodes to be added to the tree and the tree nodes to which they must be connected are determined one by one. A step-by-step implementation of the algorithm for the network example of Figure 1 is given in Table 1. The fact of choosing the node for adding it to the shortest path tree is indicated by taking it in parentheses for the first time and painting it in green.

Each step of the algorithm implementation adds a new node to the shortest paths tree (Figure 2). The first step determines the root of the tree D. In the second step of the algorithm implementation, node H is added to the tree, in the third step A, and so on. The tree of shortest paths (topology of shortest routes) for node D to all other nodes is shown in Figure 2.

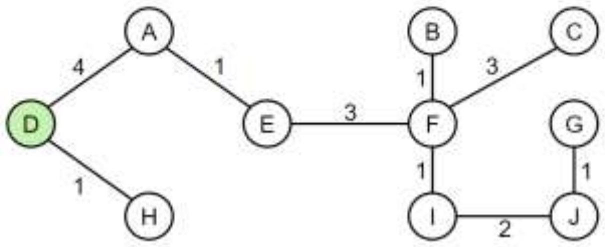

Fig. 2: The shortest path tree for node D to all other nodes according to Dijkstra's algorithm for the network is in Figure 1.

The constructed tree of the shortest paths depends only on one criterion - the weight of the channels. All nodes in it are equal from the point of view of priority inclusion in the tree. This algorithm (Dijkstra's algorithm) is the basis of our method when dealing with normal nodes, but when considering an untrusted node, its inclusion in the tree is done differently. The following section presents various scenarios that can occur in a network with an untrusted node and suggestions for how to respond to them.

4 Modifications and Results

4.1 Scenario A – There is a Distrustful Node in the Network and There are Paths to All other Nodes without Its Participation

Let node E (Figure. 3) of the network under study become the node through which the data flows transmitted to other nodes should be stopped. At the same time, packets for which node E is the destination must be delivered. Note the condition of packet transmission to all other nodes without using node E as an intermediate node is fulfilled

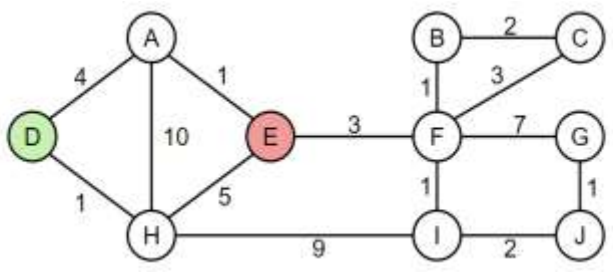

Fig. 3: The network with distrustful node E, the absence of which allows node D to transmit packets to all network nodes.

Table 1. Step-by-step implementation of Dijkstra's algorithm for the network example of Figure 1.

	Set	Channel metric of node D with nodes, DS(v)								
Step	N	A	B	C	E	F	G	H	I	J
1	{D}	4	∞	∞	∞	∞	∞	(1)	∞	∞
2	{D,H}	(4)	∞	∞	6	∞	∞	(1)	10	∞
3	{D,H,A}	(4)	∞	∞	(5)	∞	∞	(1)	10	∞
4	{D,H,A,E}	(4)	∞	∞	(5)	(8)	∞	(1)	10	∞
5	{D,H,A,E,F}	(4)	(9)	11	(5)	(8)	15	(1)	9	∞
6	{D,H,A,E,F,B}	(4)	(9)	11	(5)	(8)	15	(1)	(9)	∞
7	{D,H,A,E,F,B,I}	(4)	(9)	(11)	(5)	(8)	15	(1)	(9)	11
8	{D,H,A,E,F,B,I,C}	(4)	(9)	(11)	(5)	(8)	15	(1)	(9)	(11)
9	{D,H,A,E,F,B,I,C,J}	(4)	(9)	(11)	(5)	(8)	(12)	(1)	(9)	(11)
10	{D,H,A,E,F,B,I,C,J,G}	(4)	(9)	(11)	(5)	(8)	(12)	(1)	(9)	(11)

We start building the shortest paths tree with the classic Dijkstra algorithm. We continue it until node E, which needs to be removed from flow retranslation, is included in the paths tree. The first three rows of Table 2 illustrate this and are the same as in Table 1. It's an invariable part of Dijkstra's algorithm realization.

In step 3 of Table 2, it is determined that the running node added to the tree is distrustful node E, through which the flow of transit data must be stopped. Step 4 illustrates its inclusion in set N.

Therefore, the distrustful node E will be included in the tree and the delivery of packets will be ensured to it. To prevent the transmission of transit flows over a distrustful node it should not have outgoing channels. Since node E is distrustful, it should have no output channels in the tree, in our case, it's the painted-in-yellow EF channel. Thus, in the modified algorithm, we return to reanalyzing the connections on the node included in the tree in the previous iteration (in step 3 of Table 2). When reanalysis, we exclude from consideration the channels connecting the distrustful node with neighboring nodes that still need to be included in the tree of shortest paths. In our case, it's the EF channel with a weight of 3. Step 5 illustrates this in Table 2. Next, we work according to the classic algorithm. That is, the next node to be added to the tree will be node I, and it must be connected to node H, for which a weight of 10 was entered in Table 1.

The complete tree of the paths (Figure 4) is obtained after all nodes of the network have passed into the set N.

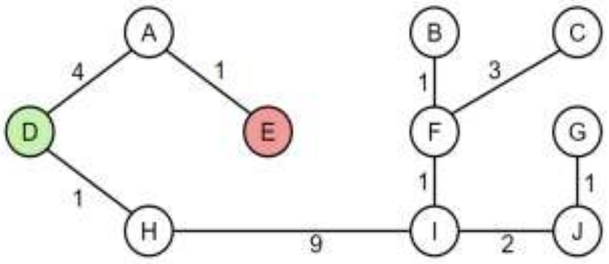

Fig. 4: The shortest path tree for node D to all other nodes according to Dijkstra's algorithm for the network is in Figure 3.

The modified algorithm can be presented as follows:

1. Form a set of nodes N, which contains the root node in the first step. N = {D}.
2. Set DS(v) = c(D,v), for all nodes v that are neighbors to node D.
3. Find node w that is not in the set N and for which DS(w) is minimum, and add node w to the set N.
4. Update DS(v) for all nodes that are not in the set N. DS(v) = min{DS(v),DS(w)+c(w,v)}.

5. If node w is distrustful, remove w from the set N. For all neighboring nodes of node w, which were assigned a numerical value in the previous step, set DS(v) = ∞.
6. Repeat steps 3-5 until all nodes are in the set N. End.

Note that the first four steps are the same as in the classical Dijkstra algorithm.

4.2 Scenario B – There are One or More Nodes Neighboring a Distrustful Node and There are No Other Paths to These Nodes Except Ways in which the Distrustful Node Must be Transitive

Let's divide all the nodes into three classes:

– trust nodes (TN);
– distrustful nodes (DN);
– post-distrustful node (PDN) - node to which packets were transmitted through the neighbor distrustful node.

Table 2. Step-by-step implementation of modified Dijkstra's algorithm for the network example of Figure 2.

	Set	Connection Metric of node D with other nodes, DS(v)								
Step	N	A	B	C	E	F	G	H	I	J
1	{D}	∞	∞	∞	∞	∞	∞	(1)	∞	∞
2	{D,H}	(4)	∞	∞	∞	∞	∞	(1)	10	∞
3	{D,H,A}	(4)	∞	∞	(5)	∞	∞	(1)	10	∞
4	{D,H,A,E}	(4)	∞	∞	(5)	∞	∞	(1)	10	∞
5	{D,H,A}	(4)	∞	∞	(5)	∞	∞	(1)	(10)	∞
6	{D,H,A,I}	(4)	∞	∞	(5)	(11)	-	(1)	(10)	12
7	{D,H,A,I,F}	(4)	(12)	14	(5)	(11)	18	(1)	(10)	12
8	{D,H,A,I,F,B}	(4)	(12)	14	(5)	(11)	18	(1)	(10)	(12)
9	{D,H.A,I,F,B,J}	(4)	(12)	14	(5)	(11)	(13)	(1)	(10)	(12)
10	{D,H,A,I,F,B,J,G}	(4)	(12)	(14)	(5)	(11)	(13)	(1)	(10)	(12)
11	{D,H,A,I,F,B,J,G,C}	(4)	(12)	(14)	(5)	(11)	(13)	(1)	(10)	(12)

We proceed with adding a post-distrustful node K (Figure 5) to our network under investigation. There is only one possible way to transmit packets to node K: through distrustful node E.

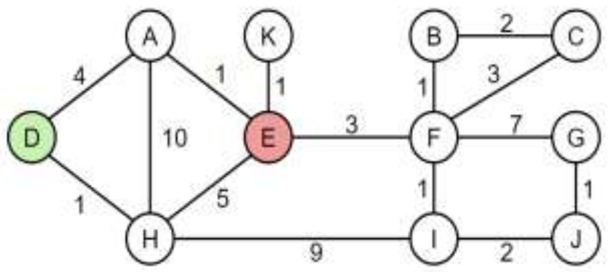

Fig. 5: The network with distrustful transit node E and post-distrustful node K.

So, in our example, node E is a distrustful node (DN), node K is a post-distrustful node (PDN), and all other nodes are trusted nodes (TN). Let's go ahead and show in Table 3 the step-by-step process of obtaining the short path tree using the modified algorithm proposed above. This process corresponds to steps 1 to 11.

In step 10, the last node with numerical value DS(v) (node C) is selected for the shortest path tree; in step 11, it will be moved to set {N}. Therefore, all TNs and distrustful node E are already included in the shortest path tree. But the post-distrustful node K remains unconnected to the tree. In step 12, this node is selected to connect to the shortest paths tree based on the data from step 4.

Now the modified algorithm can be presented as follows:

1. Form a set of nodes N, which contains the root node in the first step. N = {D}.
2. Set DS(v) = c(D,v), for all nodes v that are neighbors to node D.
3. Find node w that is not in the set N and for which DS(w) is minimum, and add node w to the set N.
4. Update DS(v) for all nodes that are not in the set N.DS(v) = min{DS(v),DS(w)+c(w,v)}.
5. If node w is distrustful, remove w from the set N. For all neighboring nodes of node w, which were assigned a numerical value in the previous step, set DS(v) = ∞.
6. Repeat steps 3-5 until all nodes with numerical value DS(v) are in the set N.

7. If not all nodes are in the set N, connect the post-distrustful nodes to the distrustful node. Otherwise, end.

The shortest paths tree is obtained and shown in Figure 6. A case with two post-distrustful nodes K and L is presented in Figure 7.

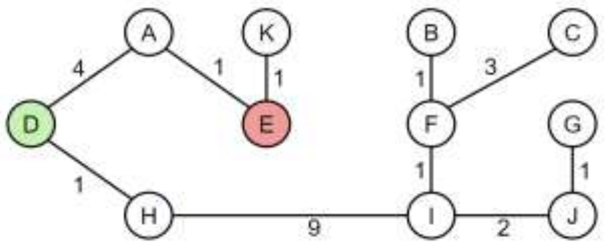

Fig. 6: The shortest paths tree for node D to all other nodes according to modified Dijkstra's algorithm for the network in Figure 5.

Table 3. Step-by-step implementation of modified Dijkstra's algorithm for the network example of Figure 6.

	Set	Connection Metric of node D with other nodes, DS(v)									
Step	N	A	B	C	E	F	G	H	I	J	K
1	{D}	4	∞	∞	∞	∞	∞	(1)	∞	∞	∞
2	{D,H}	(4)	∞	∞	6	∞	∞	(1)	10	∞	∞
3	{D,H,A}	(4)	∞	∞	(5)	∞	∞	(1)	10	∞	∞
4	{D,H,A,E}	(4)	∞	∞	(5)	8	∞	(1)	10	∞	6
5	{D,H,A}	(4)	∞	∞	(5)	∞	-	(1)	(10)	-	∞
6	{D,H,A,I}	(4)	∞	∞	(5)	(11)	-	(1)	(10)	12	∞
7	{D,H,A,I,F}	(4)	(12)	14	(5)	(11)	18	(1)	(10)	12	∞
8	{D,H,A,I,F,B}	(4)	(12)	14	(5)	(11)	18	(1)	(10)	(12)	∞
9	{D,H,A,I,F,B,J}	(4)	(12)	14	(5)	(11)	(13)	(1)	(10)	(12)	∞
10	{D,H.A,I,F,B,J,G}	(4)	(12)	(14)	(5)	(11)	(13)	(1)	(10)	(12)	∞
11	{D,H,A,I,F,B,J,G,C}	(4)	(12)	(14)	(5)	(11)	(13)	(1)	(10)	(12)	∞
12	{D,H,A,E,K}	Not in use									(6)

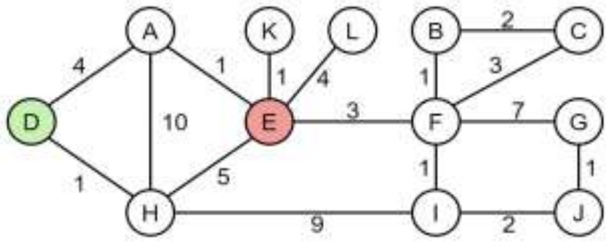

Fig. 7: The network with a distrustful node E and multiple paths in which node E must be transitive.

Table 4 shows the implementation of the algorithm given above. The shortest paths tree differs from the tree in Figure 6 only in that the additional node L is connected to node E.

4.3 Scenario C – There is a Subnet to Which Data Cannot be Transmitted without Using a Distrustful Node E

Let's go to the network shown in Figure 1 and add to it three more nodes: K, L, and M (Figure 8). There are no other links to connect nodes K, L, and M to the core network; only through the distrustful node E.

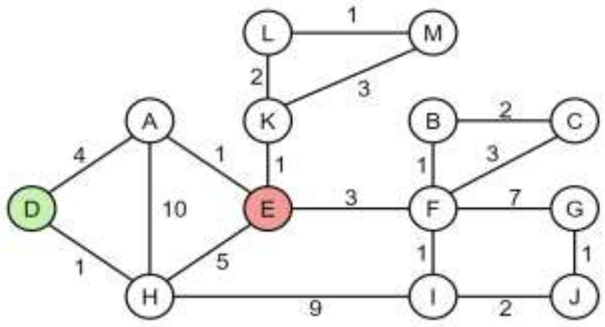

Fig. 8: A network with the distrustful transit node E to which a subnet (nodes K, L, and M) is connected.

Now we must add to the modified algorithm a part that will be able to complete the tree further from the post-distrustful node K. We must add to the common tree all nodes that have not yet been entered. We will start searching for the tree of the shortest paths according to the last modified algorithm described above.

A step-by-step illustration of this is shown in Table 5.

At iteration 11 we will see that there are nodes that did not enter the shortest paths tree. Now we return to the iteration at which the distrustful node was added to the set N. In our case, it's node E in iteration 4. Now, using the classical Dijkstra algorithm, we transfer the remaining nodes to the set N, and therefore to the shortest paths tree. Iterations 13 – 15 in Table 5 illustrate this part of the algorithm.

The shortest paths tree for Scenario C is obtained and shown in Figure 9.

The following steps can represent the final algorithmic implementation of the method to minimize transit flows through a distrustful network router.

1. Form a set of nodes N, which contains the root node in the first step. N = {D}.
2. Set DS(v) = c(D,v), for all nodes v that are neighbors to node D.
3. Find node w that is not in the set N and for which DS(w) is minimum, and add node w to the set N.
4. Update DS(v) for all nodes that are not in the set N.
5. DS(v) = min{DS(v), DS(w)+c(w,v)}.
6. If node w is distrustful, remove w from the set N. For all neighboring nodes of node w, which were assigned a numerical value in the previous step, set DS(v) = ∞.
7. Repeat steps 3-5 until all nodes with numerical value DS(v) are in the set N.
8. If not all nodes are in the set N, add the distrustful node to the set N, and continue until termination by the classical Dijkstra algorithm. Otherwise, ends.

Table 4. Step-by-step implementation of modified Dijkstra's algorithm for the network example of Figure 7.

	Set	Connection Metric of node D with other nodes, DS(v)										
Step	N	A	B	C	E	F	G	H	I	J	K	L
1	{D}	4	∞	∞	∞	∞	∞	(1)	∞	∞	∞	∞
2	{D,H}	(4)	∞	∞	6	∞	∞	(1)	10	∞	∞	∞
3	{D,H,A}	(4)	∞	∞	(5)	∞	∞	(1)	10	∞	∞	∞
4	{D,H,A,E}	(4)	∞	∞	(5)	8	∞	(1)	10	∞	6	9
5	{D,H,A}	(4)	∞	∞	(5)	∞	-	(1)	(10)	-	∞	∞
6	{D,H,A,I}	(4)	∞	∞	(5)	(11)	-	(1)	(10)	12	∞	∞
7	{D,H,A,I,F}	(4)	(12)	14	(5)	(11)	18	(1)	(10)	12	∞	∞
8	{D,H,A,I,F,B}	(4)	(12)	14	(5)	(11)	18	(1)	(10)	(12)	∞	∞
9	{D,H,A,I,F,B,J}	(4)	(12)	14	(5)	(11)	(13)	(1)	(10)	(12)	∞	∞
10	{D,H.A,I,F,B,J,G}	(4)	(12)	(14)	(5)	(11)	(13)	(1)	(10)	(12)	∞	∞
11	{D,H,A,I,F,B,J,G,C}	(4)	(12)	(14)	(5)	(11)	(13)	(1)	(10)	(12)	∞	∞
12	{D,H,A,E,K}	Not in use									(6)	9
13	{D,H,A,E,K,L}	Not in use									(6)	(9)

Table 5. Step-by-step implementation of modified Dijkstra's algorithm for the network example of Figure 8.

	Set	Connection Metric of node D with other nodes, DS(v)											
Step	N	A	B	C	E	F	G	H	I	J	K	L	M
1	{D}	4	∞	∞	∞	∞	∞	(1)	∞	∞	∞	∞	∞
2	{D,H}	(4)	∞	∞	6	∞	∞	(1)	10	∞	∞	∞	∞
3	{D,H,A}	(4)	∞	∞	(5)	∞	∞	(1)	10	∞	∞	∞	∞
4	{D,H,A,E}	(4)	∞	∞	(5)	8	∞	(1)	10	∞	6	∞	∞
5	{D,H,A}	(4)	∞	∞	(5)	∞	∞	(1)	(10)	∞	∞	∞	∞
6	{D,H,A,I}	(4)	∞	∞	(5)	(11)	∞	(1)	(10)	12	∞	∞	∞
7	{D,H.A,I,F}	(4)	(12)	14	(5)	(11)	18	(1)	(10)	12	∞	∞	∞
8	{D,H,A,I,F,B}	(4)	(12)	14	(5)	(11)	18	(1)	(10)	(12)	∞	∞	∞
9	{D,H,A,I,F,B,J}	(4)	(12)	14	(5)	(11)	(13)	(1)	(10)	(12)	∞	∞	∞
10	{D,H,A,I,F,B,J,G}	(4)	(12)	(14)	(5)	(11)	(13)	(1)	(10)	(12)	∞	∞	∞
11	{D,H,A,I,F,B,J,G,C}	(4)	(12)	(14)	(5)	(11)	(13)	(1)	(10)	(12)	∞	∞	∞
12	{TNs,E}	Not in use									(6)	∞	∞
13	{TNs,E,K}	Not in use									(6)	(8)	9

14	{TNs,E,K,L}	Not in use	(6)	(8)	(9)
15	{TNs,E,K,L,M}	Not in use	(6)	(8)	(9)

5 Discussion

In a network operating under the OSPF protocol, packets are routed in accordance with the shortest paths tree found by Dijkstra's algorithm. An example of the network chosen by us for illustration is shown in Figure 1, the step-by-step implementation of Dijkstra's algorithm in Table 1, and the found tree of the shortest paths is shown in Figure 2. Analyzing this tree, you can see that node A will be selected as a gateway for the node D routing table when transmitting data to nodes other than H.

Research concerns the case when, for some reason, trust in a specific node is lost. For example, there is a suspicion of unauthorized interception possibility of data on it.

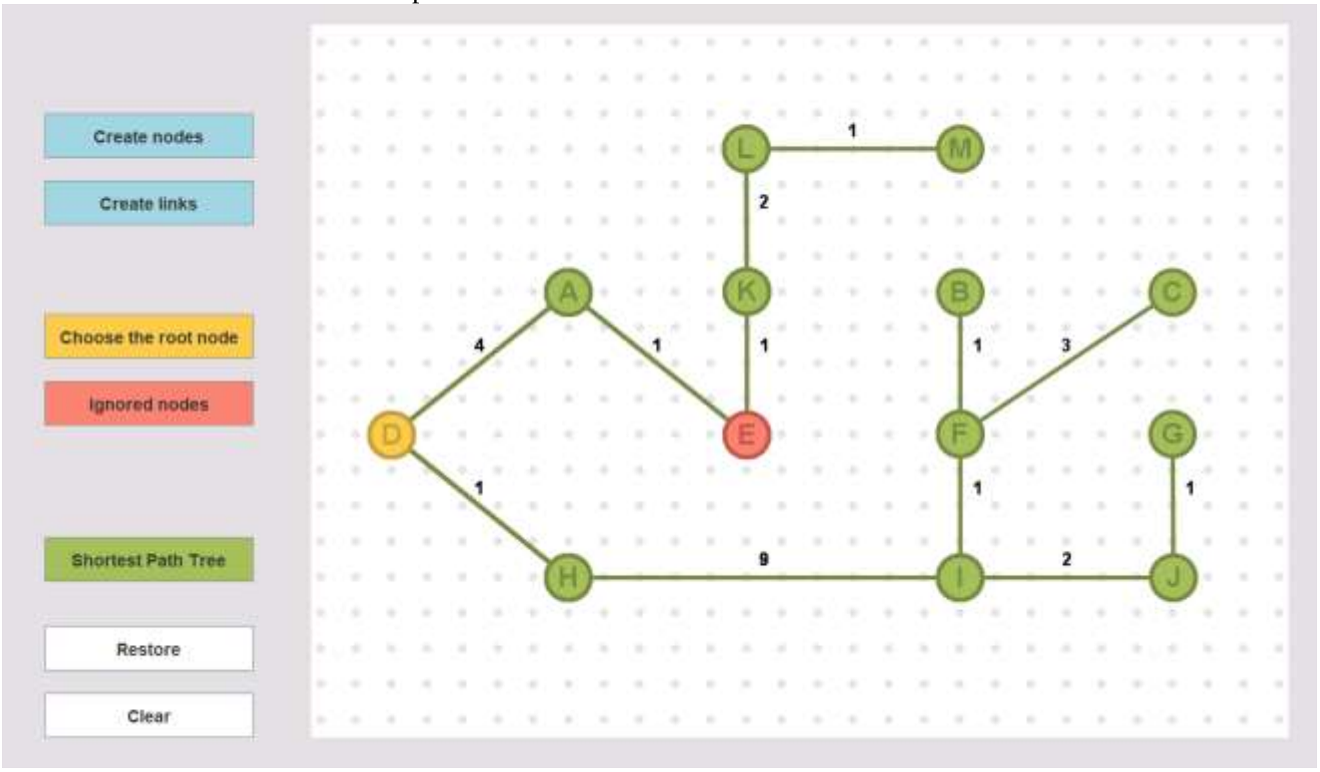

Fig. 9: The shortest paths tree for node D according to modified Dijkstra's algorithm for the network in Figure 8 with distrustful node E.

Our goal was to keep sending data to this node, but stop, if possible, sending data through it to all other nodes, or at least minimize such flows. To do this, it is suggested not to use, or if this is not possible, to minimize using the outgoing channels of the distrustful node.

In the beginning, we considered the case where there are paths to all nodes in the network without using the distrustful node as a transit. An example of the network is shown in Figure 3, a step-by-step implementation of the algorithm proposed for this case in the table. 2, and the resulting tree of shortest paths is shown in Figure 4. Comparing the trees in Figure 2 and Figure 4, we see that the appearance of a distrustful router E in the network will lead to a change in the routing table already on the root router D. If, in the absence of the distrustful router E, router A served as a gateway to almost all other networks, now, even for the most distant networks, it is an H router.

Next, we considered the cases when there are routers that have a connection only to distrustful router E. An example of the network is shown in Figure 5 and Figure 7, a step-by-step implementation of the algorithm proposed for these cases in Table 3, and Table 4 relatively, and the resulting tree of shortest paths for the network in Figure 5 is shown in Figure 6.

The last case involves the connection of a subnet to a distrustful node (the network in Figure 8) is generalized, while the previous ones are partial. The proposed algorithm allows finding a tree of the shortest paths (Figure 9), which will minimize the weight of the paths that will pass through the distrustful node in cases where there are no routes without its use.

The minimization of transit traffic through a node that has lost trust for some reason, combined with the delivery of packets addressed to it, is the main advantage of the proposed method compared to others.

6 Conclusions

In practice, for various reasons, there may be recommendations not to use a certain node as a transit, that is, not to use it to transmit data to other nodes through it. One of these reasons may be the loss of trust in this node, so in the work, we called this node distrustful. At the same time, packets to this node must be delivered in full. Our proposed short path tree search method for constructing routing tables of routers allows us to build a path tree that prevents the use of a distrustful node as a transit node if such a tree exists. Only those nodes that cannot be connected to the tree without using the distrustful node will be connected to the tree through it, but in this case, the flows through the distrustful node will be minimized.

The limitation of this study is taking into account only one criterion, as it is in the classical Dijkstra algorithm. This criterion in this paper is the weight of the channel. Future development of this research may focus on other criteria, in particular, the number of intermediate hops between the source and the remote destination.

References:

[1] Kaur, G., and Thakur, P. “Routing Protocols in MANET: An Overview.” 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), IEEE, 2019, pp. 935–41. https://doi.org/10.1109/ICICICT46008.2019.8993294.

[2] Greguš, M., Liskevych, O., Obelovska, K., and Panchyshyn, R. "Packet Routing Based on Integral Normalized Criterion," 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud), Istanbul, Turkey, 2019, pp. 393–96. https://doi.org/10.1109/FiCloud.2019.00064.

[3] Da Silva, J. M., Ramos, G. D. O., and Barbosa, J. L. V. “Multi-Objective Decision-Making Meets Dynamic Shortest Path: Challenges and Prospects.” Algorithms, vol. 16, no. 3, Mar. 2023, p. 162. https://doi.org/10.3390/a16030162.

[4] Kovtun, V., Izonin, I., Gregus, M. Model of functioning of the centralized wireless information ecosystem focused on multimedia streaming, Egyptian Informatics Journal, Volume 23, Issue 4, 2022, pp. 89-96. ISSN: 1110-8665. https://doi.org/10.1016/j.eij.2022.06.009.

[5] Broumi, S., Talea, M., Bakali, A., Smarandache, F., Nagarajan, D., Lathamaheswari, M., and Parimala, M. “Shortest Path Problem in Fuzzy, Intuitionistic Fuzzy and Neutrosophic Environment: An Overview.” Complex & Intelligent Systems, vol. 5, no. 4, Dec. 2019, pp. 371–78. https://doi.org/10.1007/s40747-019-0098-z.

[6] Drozdek, A. Data Structures and Algorithms in Java, 2nd ed., Cengage Learning, US, 2010, pp. 383-393.

[7] Madyatmadja, E. D., Nindito H., Bhaskoro, R. A., Sano, A. V. D., And Sianipar, C. P. M. “Algorithm to Find Tourism Place Shortest Route: A Systematic Literature Review.” Journal of Theoretical and Applied Information Technology, vol. 99, no. 4, 2021.

[8] Vaishali, K., Lakra, M., Sahni, R., and Verma, R. “Shortest Path Algorithms: Comparison and Applications.” A Multidisciplinary Research Journal, vol. 1, 2022, pp. 28–58.

[9] Hamela, K. “A Graph Theory Algorithm To Find Shortest Path In Routing Protocol.” International Journal of Scientific & Engineering Research, vol. 8, no. 5, May 2017, ISSN: 2229-5518.

[10] Rego, A., Sendra, S., Jimenez, J. M., and Lloret, J. “Dynamic Metric OSPF-Based Routing Protocol for Software Defined Networks.” Cluster Computing, vol. 22, no. 3, Sept. 2019, pp. 705–20, https://doi.org/10.1007/s10586-018-2875-7.

[11] Aulia, M. A., Sukmandhani, A. A., and Ohliati, J. “RIP and OSPF Routing Protocol Analysis on Defined Network Software.” 2022 International Electronics Symposium (IES), IEEE, 2022, pp. 393–97, https://doi.org/10.1109/IES55876.2022.9888355.

[12] Mohanty, S. N., Lydia, E. L., Elhoseny, M., Al Otaibi, M. M. G., and Shankar, K. “Deep Learning with LSTM Based Distributed Data Mining Model for Energy Efficient Wireless Sensor Networks.” Physical Communication, vol. 40, June 2020, p. 101097,

https://doi.org/10.1016/j.phycom.2020.101097.

[13] Kabir, M. H., Kabir, M. A., Islam, M. S., Mortuza, M. G., and Mohiuddin, M. “Performance Analysis of Mesh Based Enterprise Network Using RIP, EIGRP and OSPF Routing Protocols.” The 8th International Electronic Conference on Sensors and Applications, MDPI, 2021, p. 47, https://doi.org/10.3390/ecsa-8-11285.

[14] Ravikumar, D., Prasath, M., Kumar, N. U., Devi, V., and Vijayalakshmi, P. Internet Security Protocol for Secure Data Transmission Using OSPF and BGP. 2022, p. 020011, https://doi.org/10.1063/5.0080354.

[15] Almotiri, S. H. “Integrated Fuzzy Based Computational Mechanism for the Selection of Effective Malicious Traffic Detection Approach.” IEEE Access, vol. 9, 2021, pp. 10751–64, https://doi.org/10.1109/ACCESS.2021.3050420.

[16] Thai, C., Bao, V. N. Q., and Thai, U. H. T. “Security for Multi-Hop Communication of Two-Tier Wireless Networks with Different Trust Degrees.” REV Journal on Electronics and Communications, vol. 12, no. 3–4, Feb. 2023. https://doi.org/10.21553/rev-jec.319.

[17] Gong, L., Wang, C., Yang, H., Li, Z., and Zhao, Z. “Fine-Grained Trust-Based Routing Algorithm for Wireless Sensor Networks.” Mobile Networks and Applications, vol. 26, no. 6, Dec. 2021, pp. 2515–24. https://doi.org/10.1007/s11036-018-1106-z.

[18] Eissa, T., Razak, A. S., Khokhar, R. H., and Samian, N. “Trust-Based Routing Mechanism in MANET: Design and Implementation.” Mobile Networks and Applications, vol. 18, no. 5, Oct. 2013, pp. 666–77. https://doi.org/10.1007/s11036-011-0328-0.

[19] Sripriya, G., and T. Santha. “A Trust-Based Design for Secure and Quality of Service Routing in Mobile Ad Hoc Networks.” International Journal of Computer Networks and Applications, vol. 9, no. 5, Oct. 2022, p. 522. https://doi.org/10.22247/ijcna/2022/215913.

[20] Lazrag, H., Chehri, A., Saadane, R., and Rahmani, M. D. “A Blockchain-Based Approach for Optimal and Secure Routing in Wireless Sensor Networks and IoT.” 2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), IEEE, 2019, pp. 411–15. https://doi.org/10.1109/SITIS.2019.00072.

[21] Yang, J., He, S., Xu, Y., Chen, L., and Ren, J. “A Trusted Routing Scheme Using Blockchain and Reinforcement Learning for Wireless Sensor Networks.” Sensors, vol. 19, no. 4, Feb. 2019, p. 970. https://doi.org/10.3390/s19040970.

[22] Chen, W., Wu, H., Chen, X., and Chen, J. “A Review of Research on Privacy Protection of Internet of Vehicles Based on Blockchain.” Journal of Sensor and Actuator Networks, vol. 11, no. 4, Dec. 2022, p. 86. https://doi.org/10.3390/jsan11040086.

[23] Fatemidokht, H., Rafsanjani, M. K., Gupta, B. B., and Hsu, C. H. “Efficient and Secure Routing Protocol Based on Artificial Intelligence Algorithms With UAV-Assisted for Vehicular Ad Hoc Networks in Intelligent Transportation Systems.” IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 7, July 2021, pp. 4757–69. https://doi.org/10.1109/TITS.2020.3041746.

[24] Tsmots, I., Tkachenko, R., Teslyuk, V., Opotyak, Y., and Rabyk, V. "Hardware Components for Nonlinear Neuro-like Data Protection in Mobile Smart Systems," 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), IEEE, 2022, pp. 198–202. https://doi.org/10.1109/CSIT56902.2022.10000636.

[25] Wu, Y. J., Hwang, P. C., Hwang, W. S., and Cheng, M. H. “Artificial Intelligence Enabled Routing in Software Defined Networking.” Applied Sciences, vol. 10, no. 18, Sept. 2020, p. 6564, https://doi.org/10.3390/app10186564.

[26] Belgaum, M. R., Alansari, Z., Musa, S., Alam, M. M., and Mazliham, M. S. “Impact of Artificial Intelligence-Enabled Software-Defined Networks in Infrastructure and Operations: Trends and Challenges.” International Journal of Advanced Computer Science and Applications, vol. 12, no. 1, 2021. https://doi.org/10.14569/IJACSA.2021.0120109.

[27] Yazdinejad, A., Parizi, R. M., Dehghantanha, A., Srivastava, G., Mohan, S., and Rababah, A. M. “Cost Optimization of Secure Routing with Untrusted Devices in Software Defined Networking.” Journal of Parallel and Distributed Computing, vol. 143, Sept. 2020, pp. 36–46. https://doi.org/10.1016/j.jpdc.2020.03.021.

[28] Abbas, S., Javaid, N., Almogren, A., Gulfam, S. M., Ahmed, A., and Radwan, A. “Securing Genetic Algorithm Enabled SDN Routing for Blockchain Based Internet of Things.” IEEE Access, vol. 9, 2021, pp. 139739–54. https://doi.org/10.1109/ACCESS.2021.3118948.

[29] Srividya, P., and Devi, L. N. “An Optimal Cluster & Trusted Path for Routing Formation and Classification of Intrusion Using the Machine Learning Classification Approach in WSN.” Global Transitions Proceedings, vol. 3, no. 1, June 2022, pp. 317–25. https://doi.org/10.1016/j.gltp.2022.03.018.

[30] Liu, Q., and Liu, M. “A Multi-Hop Routing Mechanism Based on Local Competitive and Weighted Dijkstra Algorithm for Wireless Sensor Networks.” Journal of Physics: Conference Series, vol. 1621, no. 1, Aug. 2020, p. 012074. https://doi.org/10.1088/1742-6596/1621/1/012074.

[31] Khan, Z. A. Comparison of Dijkstra’s Algorithm with Other Proposed Algorithms. 2016. https://doi.org/10.13140/RG.2.2.22743.88480.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed to the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare.