computation

MDPI

Article

Minimizing the Number of Distrustful Nodes on the Path of IP Packet Transmission

Kvitoslava Obelovska *, Oleksandr Tkachuk and Yaromyr Snaichuk

Department of Automated Control Systems, Lviv Polytechnic National University, 79013 Lviv, Ukraine; oleksandr.vpro@gmail.com (O.T.); yaromyr.l.snaichuk@lpnu.ua (Y.S.)
* Correspondence: kvitoslava.m.obelovska@lpnu.ua

Abstract: One of the important directions for improving modern Wide Area Networks is efficient and secure packet routing. Efficient routing is often based on using the shortest paths, while ensuring security involves preventing the possibility of packet interception. The work is devoted to improving the security of data transmission in IP networks. A new approach is proposed to minimize the number of distrustful nodes on the path of IP packet transmission. By a distrustful node, we mean a node that works correctly in terms of hardware and software and fully implements its data transport functions, but from the point of view of its organizational subordination, we are not sure that the node will not violate security rules to prevent unauthorized access and interception of data. A distrustful node can be either a transit or an end node. To implement this approach, we modified Dijkstra's shortest path tree construction algorithm. The modified algorithm ensures that we obtain a path that will pass only through trustful nodes, if such a path exists. If there is no such path, the path will have the minimum possible number of distrustful intermediate nodes. The number of intermediate nodes in the path was used as a metric to obtain the shortest path trees. Routing tables of routers, built on the basis of trees obtained using a modified algorithm, provide increased security of data transmission, minimizing the use of distrustful nodes.

Keywords: IP networks; OSPF; packet routing; packet routing security; routing tables

Citation: Obelovska, K.; Tkachuk, O.; Snaichuk, Y. Minimizing the Number of Distrustful Nodes on the Path of IP Packet Transmission. *Computation* **2024**, *12*, 91. https://doi.org/10.3390/computation12050091

Academic Editors: Francesco Cauteruccio and Yudong Zhang

Received: 22 February 2024
Revised: 7 April 2024
Accepted: 30 April 2024
Published: 3 May 2024

1. Introduction

Routing is one of the main tasks in computer networks and its importance and impact on network efficiency are difficult to overestimate. To implement routing in networks, various routing methods, algorithms, and routing protocols are used. One of the main tasks of routing protocols is finding the shortest paths, construction, and maintenance of routing tables.

The main protocol widely used within an Internet Service Provider network is an Open Shortest Path First routing protocol (OSPF) [1]. OSPF is a dynamic routing protocol based on link-state technology, which uses Dijkstra's algorithm for computing least cost paths. The algorithm allows you to find the shortest path using the selected criterion, while not allowing for the provision of other additional requirements. In our work, as such an additional requirement, we set a goal not to include, if possible, destination routers that are not trustful. The reasons why the router will be considered distrustful may be different, for example, due to its location in the temporarily occupied territory, in the area near the fighting, or others. The responsibility for assigning a router to the untrusted class or returning it back to the normal class can be delegated only to Autonomous Systems, since they are the ones who determine the routing policy on the Internet. All routers of the Autonomous System with a unique ASN number are under the control of a single technical administration service, which clearly defines the policy of routing the IP packets within it.

Routing problems are widely covered in the world's scientific literature. These are works of a system-wide nature, which analyze, for example, static and dynamic routing

or centralized and decentralized routing [2,3], as well as works that analyze, compare, and improve specific protocols under certain conditions. In particular, many works are devoted to the implementation of the widely used OSPF protocol in Wide Area Networks (WANs) [4–9]. Different tools are used for routing research, such as the Riverbed Modeler [10], the GNS3 [7], the Cisco Packet Tracer simulator [11], etc. These make it possible to study the behavior of protocols in networks of different topologies, at different data transfer rates, to analyze various indicators using different criteria.

The most common criteria of routing in WANs are the number of routers that a packet must pass through to reach the destination network or the channel throughput [12] and, in the sensor networks, the power consumption [13]. The authors of [14] study the potential of using the relatively new metric of Age-of-Information for enhancing delay-tolerant routing protocols. Delays due to signal propagation time in the physical environment, cost, reliability, and security are also often used in various networks.

The OSPF protocol allows us to obtain the shortest paths, taking into account one of these criteria. The disadvantage of the single-criteria approach is that the optimal packet transmission path found for this criterion may not be acceptable from the point of view of other criteria, which are also important in this case. In [15], a routing protocol for wireless sensor networks using two criteria is proposed. It involves the classification of data packets into critical and normal ones and, based on this categorization, offers two delivery paths instead of one, as is usually the case. The case in which three criteria are considered is presented in other studies [5,9].

To provide security at data transmission, different approaches are proposed. For example, a neural network for real-time cryptographic data protection oriented on unmanned aerial vehicles is presented in [16], for which data protection from damage or loss in the wireless channel between it and the remote control center is critical. The study in [17] proposes the concept of calculating the info-communication system reliability indicator of operating under the probable impact of typical cyber-physical attacks. Special issues dedicated to Internet Security, Trust, and Privacy are organized [18].

In the modern world of telecommunications, the problem of routing security is becoming more and more urgent [19–24]. The ability to divert targeted traffic via routing attacks is a new threat to Internet applications [19]. Routing attacks occur in the wild and are becoming increasingly prevalent. Among them are Routing Table Poisoning, when incorrect entries are created in the routing table, and Packet Mistreating Attacks, when the router mistreats the packets after being injected with malicious code. The latter case results in network loops, denial-of-service, and congestion. This form of attack is extremely tough to detect and debug. One of the most dangerous attacks is a sinkhole attack, where some fake node advertises a fake routing update. The authors of [20] systematize modern sinkhole attack detection methods in wireless sensor networks. Both sensor networks and the Internet of Things impose additional challenges for the organization of their security. There are certain problems and influencing factors that must be taken into account when developing a protection system for data transmission in these networks. The main ones are limited computational abilities; ensuring requirements for the limited energy consumption, dimensions, and weight of the equipment; provision of data transmission in real time; ensuring the requirements for the cost; problems of encryption on regular transmissions; and a change in environmental conditions that affects the quality of data transmissions such as weather conditions, electromagnetic noise, temperature, etc. [16,25–27].

The secure and Quality of Service (QoS) routing in MANETs is presented in [23]. When constructing the path, it is taken into account whether the nodes exhibit various packet forwarding misbehaviors. It is recommended to use intelligent rules to make decisions in routing where the environment is unreliable, unpredictable, and dynamic. Another example of secure and QoS routing in MANET is given in [28]. The authors propose the use of a combined criterion of global and local trust, when choosing the next node for packet transmission. The trust measurement approach is based on information about packets that have previously been routed through the network, as well as the certain properties of a

particular node. Since the method is developed for MANETs and relies only on information from the network to classify nodes, it cannot be extended to use in WANs.

An extensive literature review on modern secure routing mechanisms for the low-powered IoT network is given in [24]. The paper analyzes the research challenges of the Internet of Things; considers various types of network attacks in WSN-IoT infrastructures; and identifies blockchain, cryptography algorithms, and artificial intelligence as the most trending WSN-IoT security methods. The failure of nodes and channels in communication networks is also a serious problem and requires fast re-routing [29].

Known versions of the modified Dijkstra algorithm are adapted to avoid certain routers or channels (nodes or edges) on the path to the destination. All of these modifications are for when these routers or links are either completely inoperable or their hardware and/or software components are malfunctioning. The innovative approach of our work consists of the fact that the distrustful nodes we consider are able to fully and correctly perform the functions of transmitting packets. However, we suspect that, in terms of their administrative support, they do not guarantee the security of the data transmitted through them. So, this is a new approach to problem formulation and, accordingly, it requires a new approach to its implementation. Common to these two approaches (known and proposed) is that, if a packet can be transmitted without the involvement of a faulty or distrustful node, it will be delivered without its involvement.

The difference between the known and the proposed approach is as follows:

- first, we must ensure the transfer of all data to all distrustful nodes (in known versions, data are not delivered to them);
- second, if there are no paths to transmit packets to some nodes, without the participation of distrustful nodes, they will still be used for transmission;
- third, the number of distrustful nodes, if they must be used, should be minimized, as well as the total number of all intermediate nodes.

In this work, we propose an approach in which potentially threatening nodes (distrustful) will not be included in the routing table as intermediate nodes, if possible. If it is impossible to have a path without distrustful nodes, their number in the route should be as minimal as possible. To some extent, it can be considered that node trust is the second additional criterion used by us when building routing tables. It should be noted that the value of this criterion (trustworthy or untrustworthy) may change over time.

Thus, the object of this study is the process of routing in telecommunication networks.

The subject of this research is algorithms used to obtain the shortest path tree, when the routing table is constructed.

This research aims to propose paths for the transmission of packets that will contain the minimum possible number of distrustful nodes.

The main contributions of this paper can be summarized as follows:

- we substantiated the need to consider an additional condition when forming routing tables in a network with distrustful nodes, for example, in the case of temporarily occupied territories;
- we modified Dijkstra's algorithm for obtaining the shortest path tree, based on the criterion of the minimum number of hops, which will provide the shortest path without intermediate distrustful nodes, if such a path exists;
- we also show that our algorithm will find a path with the minimum possible number of distrustful nodes, for the case when there is no shortest path without using such nodes.

2. Materials and Methods

2.1. Statement of Research

One of the most common routing protocols used in the Internet Autonomous System is the OSPF protocol, described in [1]. Most router manufacturers support this protocol. The protocol is based on the link state analysis algorithm. Link state analysis algorithms provide each router with enough information to construct a network's link graph. All routers work based on the same graphs, which makes the routing process stable. Routers periodically

exchange short service packets with their nearest neighbor routers. This traffic is broadcast, but it circulates only between neighbors and, therefore, clogs the network slightly.

The OSPF protocol is based on the shortest path tree search algorithm. The set of shortest paths from a certain node to all known remote networks is the shortest path tree for this node. For each node of the Autonomous System, it is necessary to find a tree of shortest paths; Dijkstra's algorithm is used to build them. Routing tables are formed based on the tree of the shortest paths. Routing tables are stored in the memory of routers and are used to select the next node (router) on the path of the packet to the destination. This node is called Gate Way (GW). Each intermediate router, after receiving a packet to be forwarded to a remote addressee, finds, in the routing table, the address of the next GW on the transmission path.

The OSPF protocol ensures the transmission of packets along the shortest paths calculated, according to a certain single criterion. Channel capacity, the number of steps to the goal, packet delivery delay, the number of datagrams in the queue for transmission, channel loading, security requirements, and others can be used as this criterion. One of the typical criteria for finding the tree of shortest paths is the number of transitive nodes on the path between the source node and the destination node, called hops. Our work will also be based on this criterion. However, there are cases when not one criterion but two or more are important for choosing paths. Accordingly, there are recommendations for the simultaneous consideration of several criteria and their various combinations. Sometimes it is important to take into account not only the values of the metrics of various criteria, but also the ratio between them. For example, in studies [5,9], the indicators of three criteria (throughput, hops, and delay) and their weight are taken into account. This task differs from the previous ones, considering one criterion, but, at the same time, an additional condition will be formulated. All nodes will be divided into two classes—trustful and distrustful—and the modified algorithm should minimize the number of untrusted nodes in the tree of shortest paths.

A weighted graph corresponding to the network topology is used as the input data for Dijkstra's algorithm. The algorithm treats each router as a node of the graph, each channel as its edge, and the weight of the edge as the value of the channel metric, a parameter used to estimate its cost.

Dijkstra's algorithm finds the shortest path tree from the source node (the root of the graph) to each destination node. The total cost of each path is calculated as the sum of the weights of the individual channels included in it. The path whose total cost is minimal is chosen for the transmission of packets.

The subject of this research is the methods used to obtain the shortest path tree for construction routers' routing table in the networks with one or more distrustful nodes.

To begin, we will formulate two requirements that must be provided using the modification of Dijkstra's algorithm, as follows:

1. It is necessary that the algorithm builds paths to bypass distrustful nodes, and the weight of the bypass path should be as minimal as possible;
2. If it is impossible to find a detour of distrustful nodes, the proposed route must pass through a minimum number of distrustful nodes.

First, consider the problem of bypassing distrustful nodes. To find a path that does not include distrustful nodes, we can simply not consider them as neighbors of the current node, using the classic Dijkstra algorithm. Note that the constructed tree of paths will not contain distrustful nodes, so the paths to them will be unknown. But the shortest path tree must include paths to them as well.

With the proposed approach, the following options are possible:

- There are workarounds to all distrust nodes and we obtain the shortest path tree using the usual Dijkstra algorithm. After that, the shortest paths to distrustful nodes must be added to the tree of shortest paths. If there is one such node, then it can be found using the classical algorithm, but if there are more nodes, it requires more detailed analysis and decision-making.

- There is no workaround for at least one distrust node and we will obtain confirmation of this. In this case, it is necessary to proceed to ensure the second requirement described above. It is necessary to supplement the tree with paths containing distrustful nodes as transit, but their number should be kept to a minimum.

There are several approaches to minimizing the number of distrust nodes on the way to the end node.

One of these approaches suggests the following additional procedure in the classical algorithm when passing the current node and counting the weight of the channel from it to all its neighbors. If you encounter a distrust node on the way, it is necessary to assign some sufficiently large positive constant to the weight of the channels associated with it, as is the same for all distrustful nodes. Thus, the inclusion of each distrustful node in the path strongly increases the running weight of the path. Note that it is the running weight that increases when passing Dijkstra's algorithm, but not the actual weight of the path, which is calculated by the sum of all weights on the edges between all vertices of the path. The larger the amount of such distrustful nodes in the path, the larger the running path weight for Dijkstra's algorithm. As a result, a path with a minimum number of distrust nodes will be found.

Another approach is to immediately look for paths to trustful and distrustful nodes, as well as either a detour or a path through the smallest number of distrustful nodes.

2.2. Modified Dijkstra's Algorithm

Let the node for which we build the shortest path tree be called the root node. Any node that should send data can be the root.

All channels between routers in the network can support bidirectional exchange. That is why each node can be the root of the classical and proposed algorithm. As a metric to obtain the shortest path trees, the number of hops in the path will be used.

The input data are a set of nodes (routers) N, a set of channels present in the network (graph arcs), a list of the distrustful nodes, and a root node.

The flowchart diagram that depicts the proposed modified approach is represented in Figure 1.

The proposed approach starts with an iterative process of assigning weight to each channel, depending on whether it is connected to a distrustful node. Channels connected to two distrustful nodes receive a weight of $2 \times M$, while those connected to one distrustful node receive a weight of M. Channels without connected distrustful nodes receive a weight value of 1. This process is repeated for all channels.

The value of M is obtained as the sum of all network channels. Therefore, the approach does not depend on the number of nodes or channels.

After the weight assignment process, the Dijkstra algorithm is executed. Then, a weight of 1 is assigned to all channel weights, and the metrics of all paths are calculated.

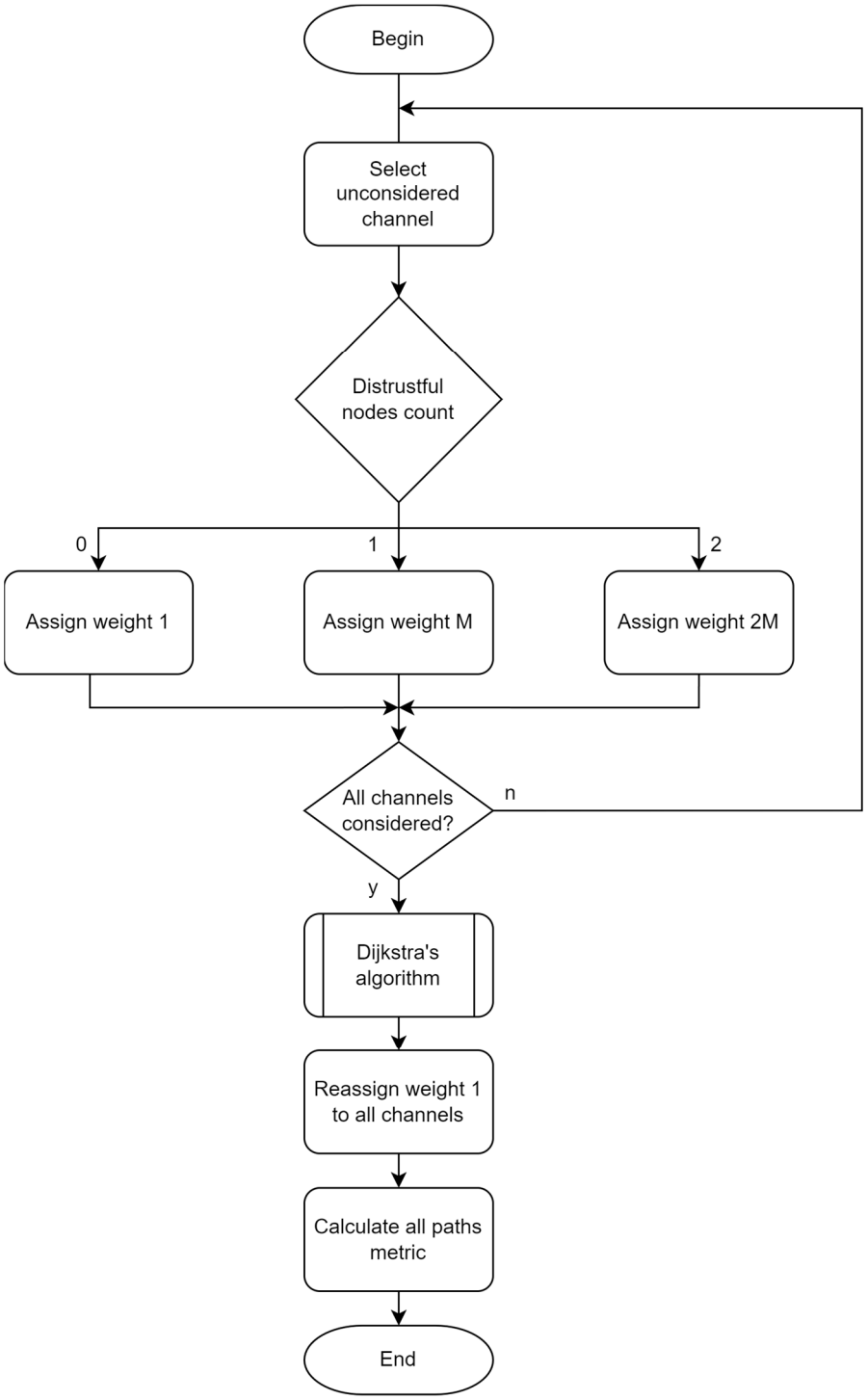


Figure 1. Flowchart diagram of modified algorithm.

3. Results and Discussion

To begin with, we will demonstrate the effectiveness of the modified algorithm using the results of the program that implements it on simple and large networks.

3.1. Scenario A, with One Distrustful Node in a Small Network

Consider the small network shown in Figure 2. Let node E be a distrustful node and node A be a router for which a shortest path tree must be constructed using the minimum hops criterion.

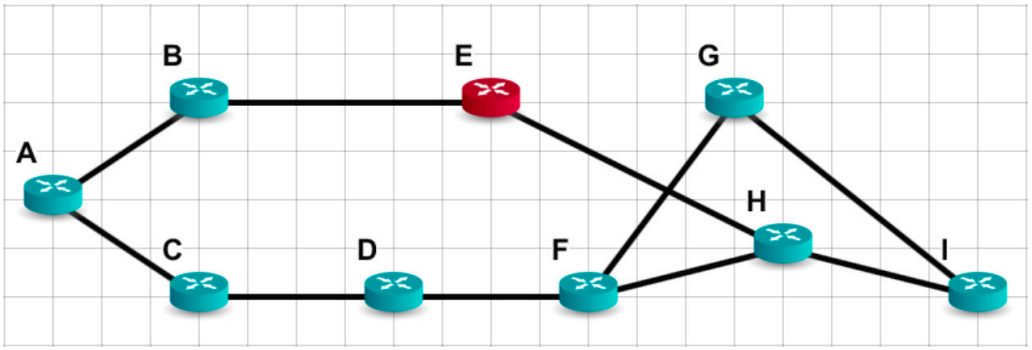


Figure 2. Network topology of a small network.

To begin with, let us illustrate how the shortest path tree for node A (Figure 3) will look if we use the classic Dijkstra algorithm. Path channels without distrustful nodes are marked in green. Path channels with distrustful nodes are marked green on the path to the first distrustful node and red after it. The gray color is used for marking channels that are not included in the shortest path tree.

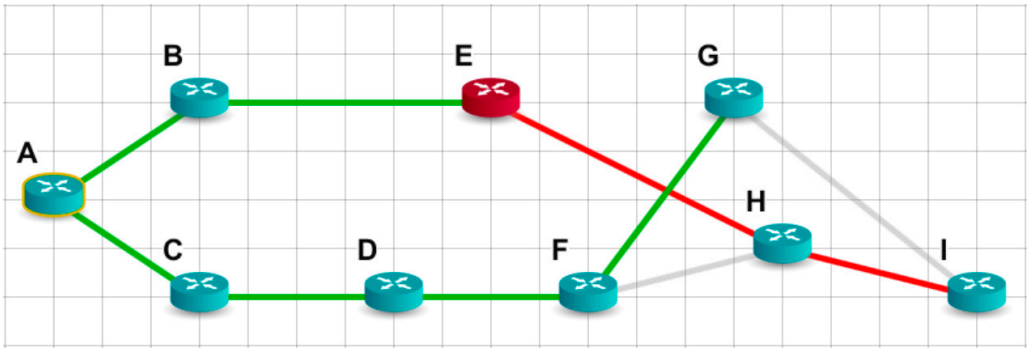


Figure 3. The shortest path tree for node A, according to classical Dijkstra's algorithm.

As can be seen from Figure 3, a shortest path tree rooted at node A has two branches. One of them has only trustful nodes and it will transmit packets from node A to nodes C, B, F, and G. The other branch has the distrustful node E in it, and the packets that will be transmitted from node A to nodes H and I will pass through this distrustful transit node.

Now, to demonstrate the efficiency of the proposed method, let us look at the result of the shortest path tree (Figure 4), obtained using the modified algorithm, and compare it with the shortest path tree of the basic Dijkstra algorithm.

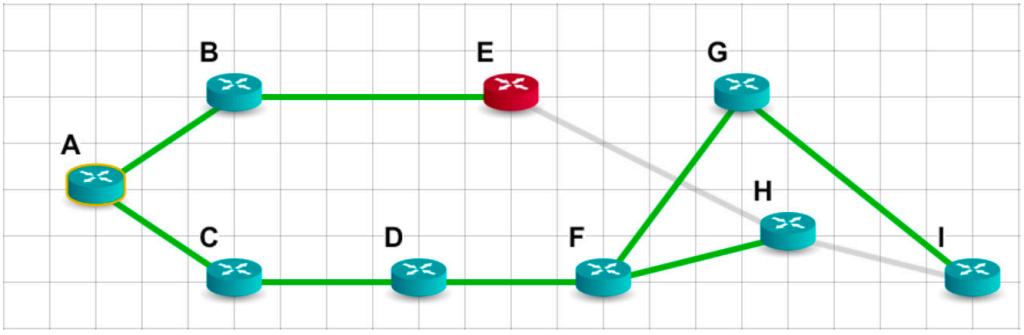


Figure 4. The shortest path tree for node A, according to the modified Dijkstra algorithm.

The figure shows that the distrustful node E is now the end of the tree branch, so it is no longer transitive. So, thanks to our approach, the task was successfully solved. Node A can transmit packets to eight nodes, including the distrustful one. The distrustful node

E will not transmit packets to any other node, because it is at the end of the branch of the shortest path tree. Note that with the classical approach, packets were transmitted to two nodes, H and I, through the distrustful node E.

The figure also shows that the paths to nodes H and I have changed. These paths are longer, but they bypass the distrustful node E. Increasing the weight of the path is the price we paid for obtaining the results of having no distrustful nodes on the path to all destination nodes.

Moreover, let us see how the data in the routing tables will change when a distrustful node E appears in the network. Figure 5 demonstrates this for Router A (node A). All reachable destinations are in the field 'Destination network'. For clarity, examples of network IP addresses are marked in parentheses next to the router name. The 'Gateway' field contains the node adjacent to Router A, through which the shortest path passes. The number of hops is shown in the field 'Metric' and the last additional column illustrates the shortest path. The left part of Figure 5 shows the case when the paths and gateways are obtained using the classic Dijkstra algorithm, while on the right is the result of the modified algorithm for the case when Router E has become distrustful.

Routing table A

Destination network	Gateway	Metric	Path
B — 221.163.79.0	B	1	A→B
C — 84.0.0.0	C	1	A→C
D — 203.85.64.0	C	2	A→C→D
E — 171.15.0.0	B	2	A→B→E
F — 208.5.240.0	C	3	A→C→D→F
G — 5.0.0.0	C	4	A→C→D→F→G
H — 172.5.0.0	B	3	A→B→E→H
I — 149.3.0.0	B	4	A→B→E→H→I

Routing table A

Destination network	Gateway	Metric	Path
B — 221.163.79.0	B	1	A→B
C — 84.0.0.0	C	1	A→C
D — 203.85.64.0	C	2	A→C→D
E — 171.15.0.0	B	2	A→B→E
F — 208.5.240.0	C	3	A→C→D→F
G — 5.0.0.0	C	4	A→C→D→F→G
H — 172.5.0.0	C	4	A→C→D→F→H
I — 149.3.0.0	C	5	A→C→D→F→G→I

Figure 5. Fragments of routing tables: on the left for the classic Dijkstra algorithm, on the right for the modified one, the changed paths are circled in red.

A comparison of the data in Figure 5 shows that, when the distrustful node E appeared, for the destination H and I, Gateway B was replaced by Gateway C. The delivery path A→B→E→H was replaced by A→C→D→F→H, and A→B→E→H→I was replaced by A→C→D→F→G→I. As we can see in the paths obtained with the modified algorithm, the distrustful node E is not used as a transit.

3.2. Scenario B, with Several Distrustful Nodes in a Small Network

Let two more nodes (D and F) in the network of Figure 2 be replaced by distrustful ones. The network now looks as is shown in Figure 6 and there are no paths from node A to nodes G, H, and I through trust nodes only.

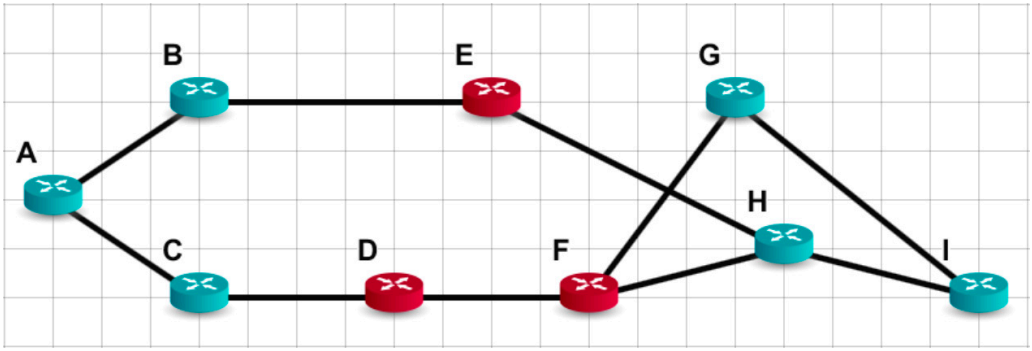


Figure 6. Network topology with several distrustful nodes.

Figure 7 illustrates the shortest path tree for node A of this network, built on the classic Dijkstra algorithm.

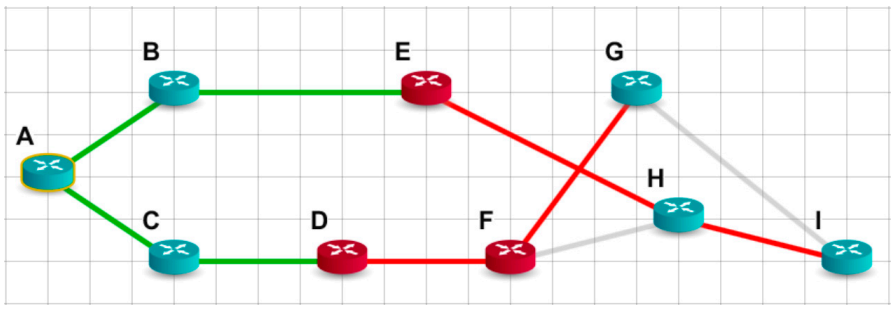


Figure 7. The shortest path tree for the network with multiple distrustful nodes, according to the classical Dijkstra algorithm.

This tree has two branches—one uses one transit distrustful node, E, but the other includes two such nodes, D and F, which are on the way to node G.

For comparison, we will build the shortest path tree according to the modified algorithm. The path to each node should contain the smallest amount of distrustful transit nodes and, at the same time, have a minimum possible weight. The result of the program is shown in Figure 8.

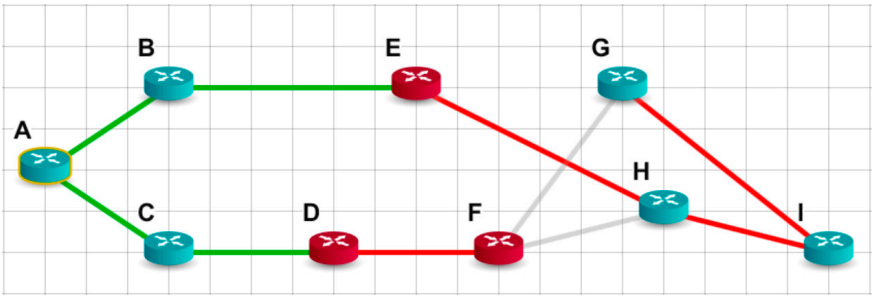


Figure 8. The shortest path tree for the network with multiple distrustful nodes, according to the modified Dijkstra algorithm.

As we can see, in the tree built according to the modified algorithm, there is a path to any node through no more than one distrustful transit node. This is node E on the way to nodes G, H, and I, and node D on the way to F. So, in this way, we demonstrated the achievement of minimizing the number of transit distrustful nodes on the way to the addressees.

After the appearance of distrustful nodes D, E, and F in the network (Figure 9), let us examine how the routing table for Node A has changed.

Routing table A

Destination network	Gateway	Metric	Path
B — 221.163.79.0	B	1	A→B
C — 84.0.0.0	C	1	A→C
D — 203.85.64.0	C	2	A→C→D
E — 171.15.0.0	B	2	A→B→E
F — 208.5.240.0	C	3	A→C→D→F
G — 172.5.0.0	C	4	A→C→D→F→G
H — 149.3.0.0	B	3	A→B→E→H
I — 141.38.0.0	B	4	A→B→E→H→I

Routing table A

Destination network	Gateway	Metric	Path
B — 221.163.79.0	B	1	A→B
C — 84.0.0.0	C	1	A→C
D — 203.85.64.0	C	2	A→C→D
E — 171.15.0.0	B	2	A→B→E
F — 208.5.240.0	C	3	A→C→D→F
G — 172.5.0.0	B	5	A→B→E→H→I→G
H — 149.3.0.0	B	3	A→B→E→H
I — 141.38.0.0	B	4	A→B→E→H→I

Figure 9. Fragments of routing tables for a network with three distrustful nodes: on the left for the classic Dijkstra algorithm, on the right for the modified one, the changed path is circled in red.

As can be seen from Figure 9, the gateway for router G has been changed from C to B. As a result, the metric and path have become longer. This change occurred because there were two distrustful nodes (D and F) on the way to node G and, after the change, there was only one distrustful node, which is E.

So, the comparison of the shortest path trees and routing tables constructed using the classical and modified algorithms illustrates the effectiveness of the modified algorithm, according to the criterion of minimizing the number of distrustful transit nodes.

To more clearly demonstrate the advantages of the proposed approach, three more examples of network topologies with an increased number of nodes are given below.

3.3. Scenario C, with Added Subnet and Two Distrustful Nodes

Let us extend the previous topology with an additional subnet, which includes nodes P, O, J, K, L, M, and N. Now, our network (Figure 10) has 16 nodes, two of them—E and G—are distrustful.

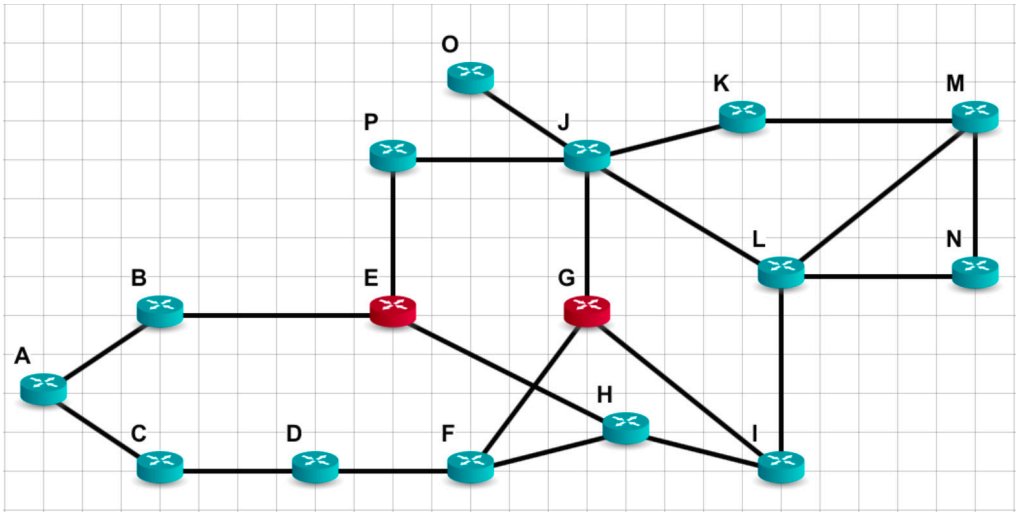
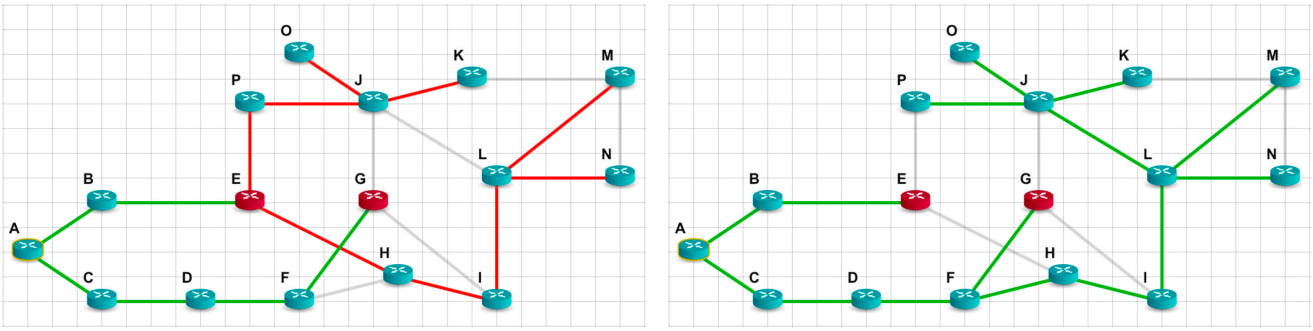


Figure 10. Network topology with 16 nodes and 2 distrustful nodes.

Let us choose node A as the root of the tree, that is, this node will correspond to the router whose routing table must be obtained. Figure 11a,b show the tree of shortest paths for the classic Dijkstra algorithm and the modified Dijkstra algorithm, respectively. Red shows the part of the path that the packets will travel, after they have been processed in a distrustful node.



(**a**) (**b**)

Figure 11. The shortest path tree for the (**a**) classical Dijkstra algorithm and (**b**) modified Dijkstra algorithm.

As shown in Figure 11, when using the classical algorithm, nine nodes (P, O, J, K, H, I, L, M, and N) will receive data through distrustful node E. When using the proposed

algorithm, we have excellent results and there are no distrustful nodes on the path to any node. We would like to note that this is not a purely theoretical case; in real networks, there may well be topology variants with such results.

The obtained results were achieved because router A, as a Gateway, chooses node C, not B, as is the case with the classic Dijkstra algorithm. Figure 12 illustrates this as a Gateway change if the Destination network is P, O, J, and K nodes.

Routing table A

Destination network	Gateway	Metric	Path
B — 221.163.79.0	B	1	A→B
C — 84.0.0.0	C	1	A→C
D — 203.85.64.0	C	2	A→C→D
E — 171.15.0.0	B	2	A→B→E
F — 208.5.240.0	C	3	A→C→D→F
G — 172.5.0.0	C	4	A→C→D→F→G
H — 149.3.0.0	B	3	A→B→E→H
I — 141.38.0.0	B	4	A→B→E→H→I
J — 9.0.0.0	B	4	A→B→E→P→J
K — 45.0.0.0	B	5	A→B→E→P→J→K
L — 157.190.0.0	B	5	A→B→E→H→I→L
M — 97.0.0.0	B	6	A→B→E→H→I→L→M
N — 146.251.0.0	B	6	A→B→E→H→I→L→N
O — 111.0.0.0	B	5	A→B→E→P→J→O
P — 170.8.0.0	B	3	A→B→E→P

(**a**)

Routing table A

Destination network	Gateway	Metric	Path
B — 221.163.79.0	B	1	A→B
C — 84.0.0.0	C	1	A→C
D — 203.85.64.0	C	2	A→C→D
E — 171.15.0.0	B	2	A→B→E
F — 208.5.240.0	C	3	A→C→D→F
G — 172.5.0.0	C	4	A→C→D→F→G
H — 149.3.0.0	C	4	A→C→D→F→H
I — 141.38.0.0	C	5	A→C→D→F→H→I
J — 9.0.0.0	C	7	A→C→D→F→H→I→L→J
K — 45.0.0.0	C	8	A→C→D→F→H→I→L→J→K
L — 157.190.0.0	C	6	A→C→D→F→H→I→L
M — 97.0.0.0	C	7	A→C→D→F→H→I→L→M
N — 146.251.0.0	C	7	A→C→D→F→H→I→L→N
O — 111.0.0.0	C	8	A→C→D→F→H→I→L→J→O
P — 170.8.0.0	C	8	A→C→D→F→H→I→L→J→P

(**b**)

Figure 12. Fragments of routing tables for the network of Figure 9, according to the (**a**) classical Dijkstra algorithm and (**b**) modified Dijkstra algorithm.

3.4. Scenario D, with Two Added Subnets and Three Distrustful Nodes

The following example (Figure 13) differs from the previous one, by adding one more subnet consisting of nodes Q, R, T, and S. Now, the total number of nodes in the network is 20, of which three are not trusted. The distrustful nodes are nodes E, G, and H.

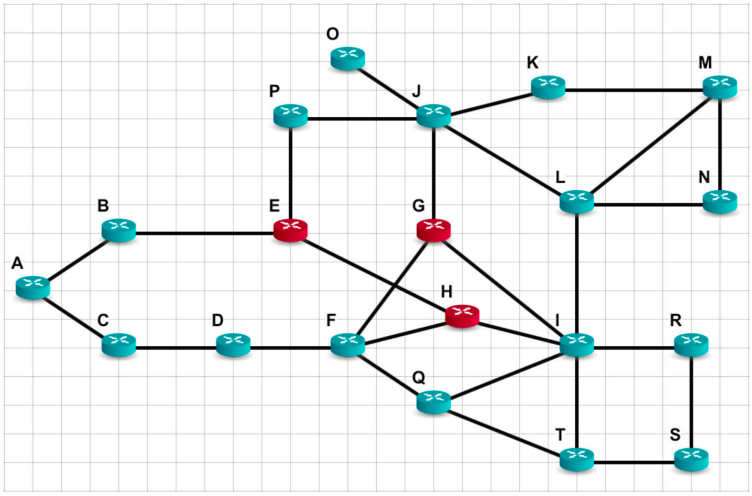


Figure 13. Network topology with 20 nodes and 3 distrustful nodes.

Similarly, as in the previous example, consider the tree of the shortest paths for the classic Dijkstra algorithm (Figure 14a) and for the modified Dijkstra algorithm (Figure 14b). The root of the tree is, again, node A.

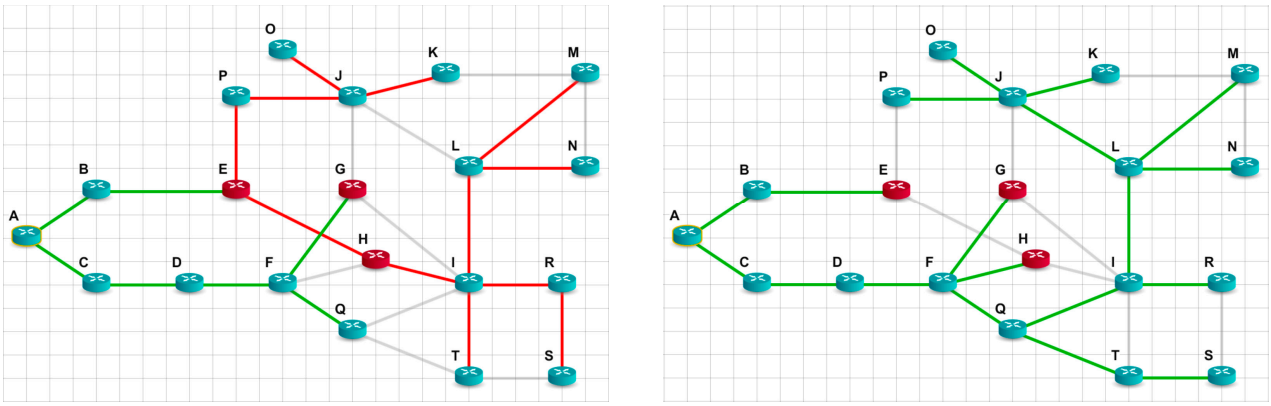


Figure 14. The shortest path tree for the (**a**) classical Dijkstra algorithm and (**b**) modified Dijkstra algorithm.

The topology of the network in Figure 13 and the location of distrustful nodes are selected in such a way as to illustrate the case of the extremely high efficiency of the proposed approach. Note that such cases can also occur in real networks. Analysis of Figure 14b shows that, when using the proposed approach, the impact of distrustful nodes E, G, and H on the security of data transmission is absent, since they will not broadcast any packets. All flows bypass these nodes. In the classical approach (Figure 14a), distrustful nodes will participate in the transport of packets up to 10 nodes (P, O, J, K, M, L, N, I, R, T, and S). Practically, only a third of the paths will not have untrusted nodes in their paths.

3.5. Large Topology Scenario

The topologies of the previous examples of networks were formed by their successive expansion and complication. In the following example, we consider a large network of arbitrary topology, the dimensions of which are chosen from the point of view of the comfortable recognition of its image. The following scenario (Figure 15) depicts the case of the presence of a distrustful node in the center of the AS and several distrustful nodes on its borders. To visually illustrate the potential possibilities of using the proposed solution, it is displayed on the background of the Ukraine map. But note that this is not an illustration of the real network. In general, it can be any Autonomous System of the Internet.

The distrustful nodes in the network are routers K, Q, R, T, U, and Y. Node K is located in the middle of the network and there are 6 nodes (G, H, J, L, O, N, and M) connected to it. Nodes Q, R, T, U, and Y are located on the periphery of the network and the number of connected nodes is relatively smaller.

Let us choose router B, for which the routing table should be constructed. So, node B will be the root of the shortest path tree that needs to be built.

To begin with, consider what the shortest path tree will look like for the classic Dijkstra algorithm (Figure 16).

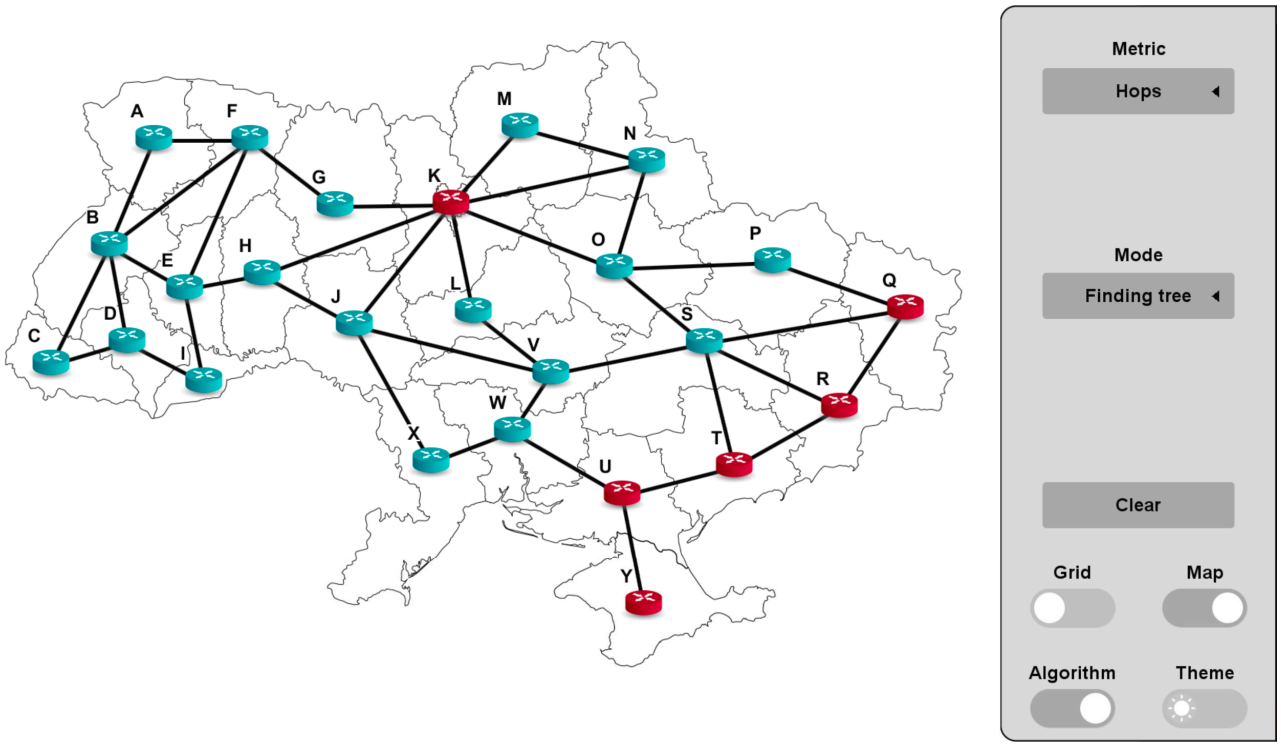


Figure 15. The topology of a large network.

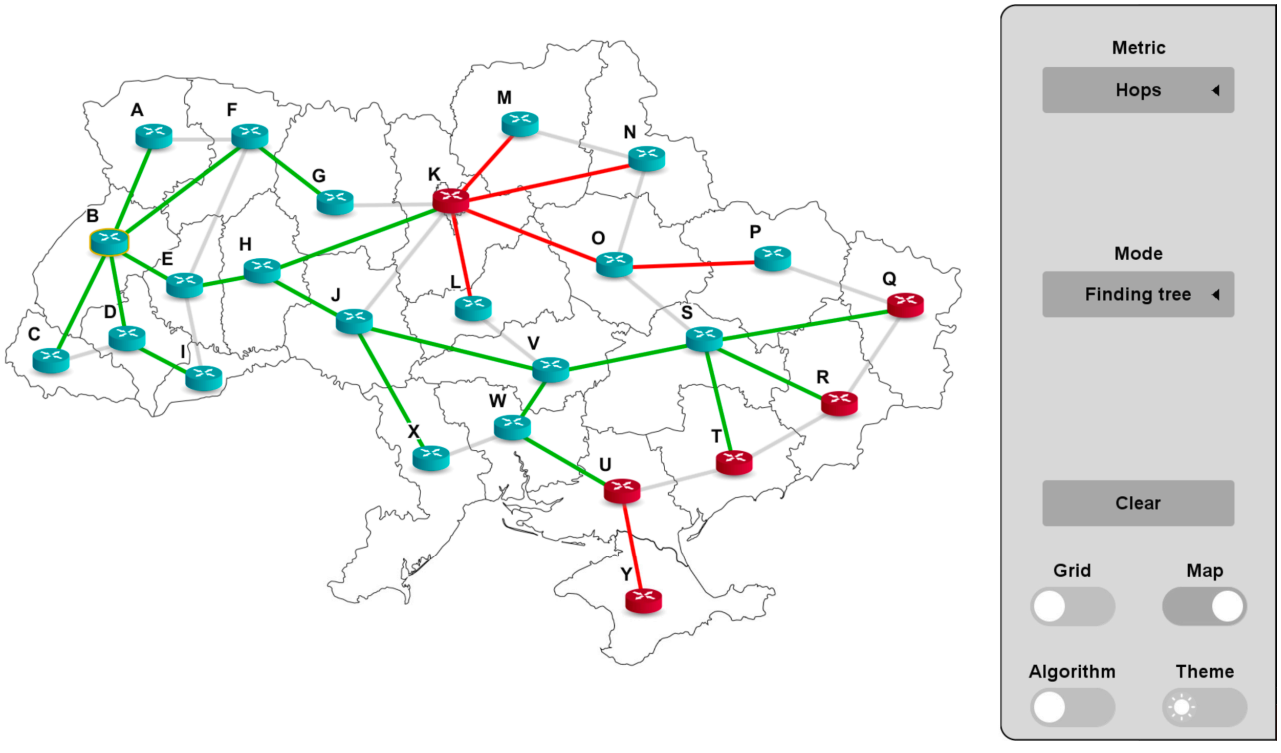


Figure 16. The shortest path tree, according to the classical Dijkstra algorithm.

Now, for comparison, consider the shortest path tree built using a modified algorithm (Figure 17).

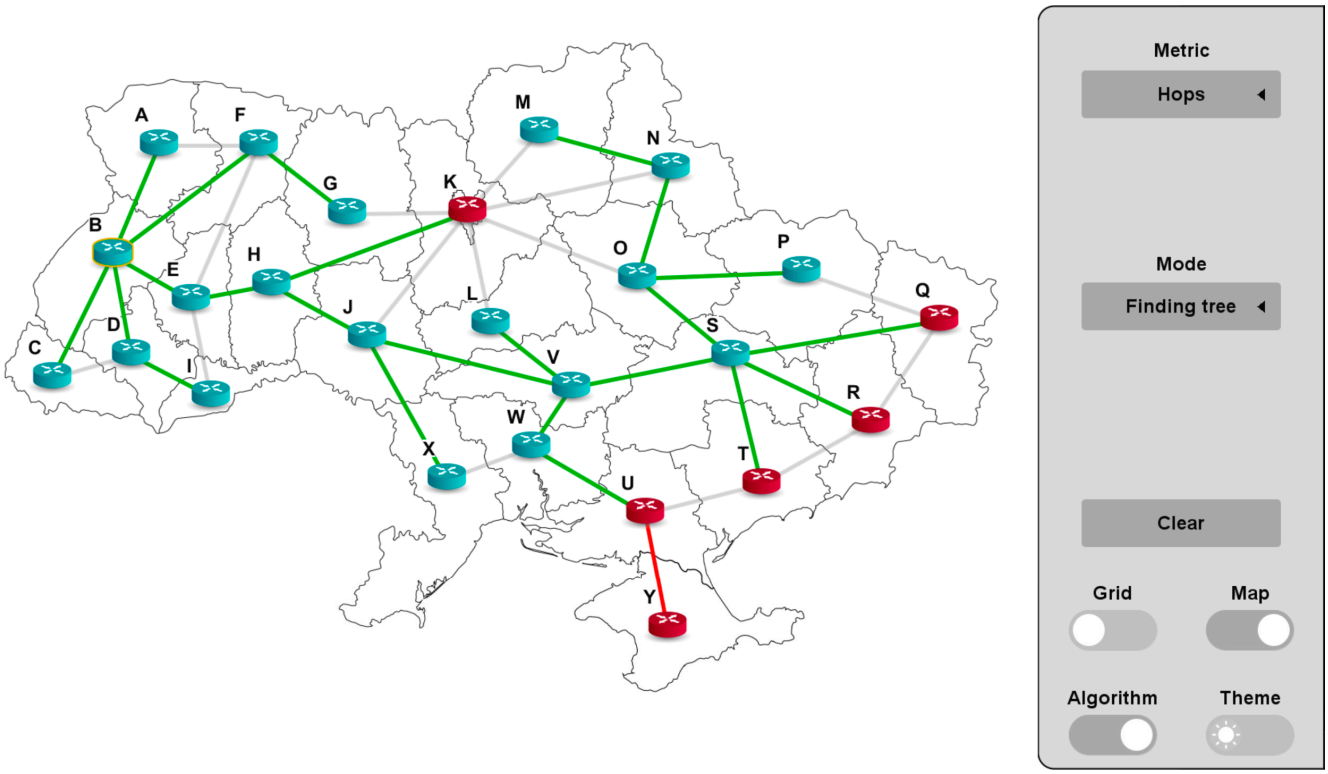


Figure 17. The shortest path tree, according to the modified Dijkstra algorithm.

As can be seen from Figure 16, the distrustful node K takes part in transporting packets to the trustful nodes M, N, O, P, and L, because the paths through it are the shortest. Distrustful nodes Q, R, and T are the end nodes of the tree branch, so there are no other paths to them. Node U is a transit node to node Y, to which there are also no bypasses.

Let us now compare the results of our modification of the algorithm (Figure 17) and Dijkstra's algorithm (Figure 16). Considering the end nodes Y, T, R, and Q, we can conclude that the paths to them have not changed, because there are no bypass paths for them. However, we can see that distrustful node K is no longer a transit node for paths to nodes M, N, O, and L. Now, the distrustful node K is an end node for a branch of this tree. Paths to nodes M, N, O, and L are now safe, but are longer than in the case of the classical algorithm. In the shown example, with the classical algorithm, the path to node M was B→E→H→K→M and was four hops long, but contained a transit distrustful node K. Our modified algorithm found a bypass for this node, which is now B→E→H→J→V→S→O→N→M and is eight hops long. In delay-tolerant networks [30], the proposed approach can be recommended without any reservations; in time-critical networks, the ratio of security and delay requirements needs to be analyzed.

The algorithm works optimally for all possible cases of the location of distrustful routers, as follows: when it is possible to bypass the distrustful node; when there is no possibility to bypass the distrustful node and it must be transitive and when distrustful routers are located both in the middle of the autonomous system and on its periphery. The algorithm works optimally regardless of how many neighboring nodes are connected to the distrustful router and what type of safety they have.

4. Conclusions

To increase packet routing security in IP networks, we have divided routers into two types. Routers of the first type have no negative warnings about their use and are included in the transmission paths on common principles. The reputation for the safe use of the second type of routers (distrustful routers) is suspect. We are unsure as to whether they will violate security rules or not, to prevent unauthorized access and the interception of

data. The reason for such a hypothesis can, for example, be related to the organizational subordination or the location of the router. We proposed that the presence of distrustful nodes, if they exist, will be taken into account when forming routing tables.

The number of hops in the path was used as a metric to obtain the shortest path trees. An algorithm has been developed that ensures obtaining a path that will pass only through trustful nodes, if such a path exists. If there is no such path, the path will have the minimum possible number of distrustful intermediate nodes. Of course, there must be a fee for this, which is an increase in the length of the path.

The proposed approach is based on the well-known and widely used computer networks Dijkstra's algorithm for constructing a tree of shortest paths. By choosing the number of hops as an optimization criterion, it was possible to solve the problem of minimizing the use of untrusted nodes for passing packets, only by changing the input data for Dijkstra's algorithm, as well as the corresponding processing of the output data, to obtain the metric of the found paths. This approach is conclusive in the correctness of the modified resulting trees, since they are constructed according to the classical Dijkstra algorithm. The given examples for different types of networks (small, medium, and large) and different locations of unreliable nodes (in the center and at the edges of the network) demonstrate the correctness of changing input and output data using the proposed approach.

The proposed approach can be used both in fully distributed mode and also in centralized mode. The centralized mode can be achieved in a WAN, managed using the principles of software-defined networking. The distributed mode is used in step-by-step routing, in which each router creates, maintains, and uses its routing table. The results of the modified algorithm indicate the potential effectiveness of its application in real Internet Autonomous Systems.

Note that the advantage of the proposed solution is the simplicity of its software implementation. This will positively affect the performance of routers when they create and rebuild routing tables during operation.

In further research, we plan to focus on improving an algorithm that will use another metric when obtaining shortest path trees.

Author Contributions: Conceptualization, K.O.; methodology, K.O. and O.T.; software, O.T.; validation, K.O., O.T. and Y.S.; formal analysis, K.O.; investigation, O.T. and Y.S.; writing—original draft preparation, K.O. and O.T.; writing—review and editing, K.O., O.T. and Y.S.; visualization, O.T.; supervision, K.O.; project administration, K.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data analyzed in this paper are in the paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Understand Open Shortest Path (OSPF)—Design Guide, [Electronic Resource]. Available online: https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/7039-1.html (accessed on 2 May 2024).
2. Aweya, J. *IP Routing Protocols: Fundamentals and Distance-Vector Routing Protocols*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2021. [CrossRef]
3. Medhi, D.; Ramasamy, K. *Network Routing Algorithms, Protocols, and Architectures, A Volume in The Morgan Kaufmann Series in Networking*, 2nd ed.; Morgan Kaufmann: Burlington, MA, USA, 2018; ISBN 9780128008294. [CrossRef]
4. Basit, Z.; Tabassum, M.; Sharma, T.; Furqan, M.; Quadir, A. Performance analysis of OSPF and EIGRP convergence through IPsec tunnel using Multi-homing BGP connection. *Mater. Today Proc.* **2022**, *62*, 4853–4861. [CrossRef]
5. Lis kevych, R.I.; Lis kevych, O.I.; Obe lovska, K.M.; Panchyshyn, R.P. Improved Algorithm for the Packet Routing in Telecommunication Networks. *Ukr. J. Inf. Technol.* **2021**, *3*, 114–119. [CrossRef]
6. Al-Musawi, B.; Branch, P.; Hassan, M.F.; Pokhrel, S.R. Identifying OSPF LSA falsification attacks through non-linear analysis. *Comput. Netw.* **2020**, *167*, 107031. [CrossRef]
7. Biradar, A.G. A Comparative Study on Routing Protocols: RIP, OSPF and EIGRP and Their Analysis Using GNS-3. In Proceedings of the 2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, India, 1–3 December 2020; pp. 1–5. [CrossRef]

8. Diansyah, T.M.; Handoko, D.; Faisal, I.; Yunianti, A.; Chiuloto, K.; Liza, R. Design Analysis of OSPF (Open Shortest Path First) Routing by Calculating Packet Loss Of Network WAN (Wide Area Network). *J. Phys. Conf. Ser.* **2019**, *1361*, 012087. [CrossRef]
9. Greguš, M.; Liskevych, O.; Obelovska, K.; Panchyshyn, R. Packet Routing Based on Integral Normalized Criterion. In Proceedings of the 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud), Istanbul, Turkey, 26–28 August 2019; pp. 393–396. [CrossRef]
10. Warsame, M.A.; Sevin, A. Comparison and Analysis of Routing Protocols Using Riverbed Modeler. *Sak. Univ. J. Sci.* **2019**, *23*, 16–21. [CrossRef]
11. Obelovska, K.; Kozak, I.; Snaichuk, Y. Analysis and Comparison of Routing and Switching Processes in Campus Area Networks Using Cisco Packet Tracer. In *Advances in Intelligent Systems, Computer Science and Digital Economics IV. CSDEIS 2022*; Hu, Z., Wang, Y., He, M., Eds.; Lecture Notes on Data Engineering and Communications Technologies; Springer: Cham, Switzerland, 2023; Volume 158. [CrossRef]
12. Obelovska, K.; Snaichuk, Y.; Selecky, J.; Liskevych, R.; Valkova, T. An Approach Toward Packet Routing in the OSPF-based Network with a Distrustful Router. *WSEAS Trans. Inf. Sci. Appl.* **2023**, *20*, 432–443. [CrossRef]
13. Krishnamoorthy, V.K.; Izonin, I.; Subramanian, S.; Shandilya, S.K.; Velayutham, S.; Munichamy, T.R.; Havryliuk, M. Energy Saving Optimization Technique-Based Routing Protocol in Mobile Ad-Hoc Network with IoT Environment. *Energies* **2023**, *16*, 1385. [CrossRef]
14. Kallitsis, G.; Karyotis, V.; Papavassiliou, S. On the Potential of Enhancing Delay-Tolerant Routing Protocols via Age of Information. *Future Internet* **2022**, *14*, 242. [CrossRef]
15. Vijayalakshmi, P.; Nguyen, T.N.; Dinakaran, J.A.; Cengiz, K. Towards sustainable energy efficient routing for dynamic ad-hoc communications in smart cities. *Measurement* **2022**, *189*, 110623. [CrossRef]
16. Tsmots, I.; Teslyuk, V.; Łukaszewicz, A.; Lukashchuk, Y.; Kazymyra, I.; Holovatyy, A.; Opotyak, Y. An Approach to the Implementation of a Neural Network for Cryptographic Protection of Data Transmission at UAV. *Drones* **2023**, *7*, 507. [CrossRef]
17. Kovtun, V.; Izonin, I.; Gregus, M. Reliability model of the security subsystem countering to the impact of typed cyber-physical attacks. *Sci. Rep.* **2022**, *12*, 12849. [CrossRef] [PubMed]
18. Meng, W.; Giannetsos, T.; Jensen, C.D. Information and Future Internet Security, Trust and Privacy. *Future Internet* **2022**, *14*, 372. [CrossRef]
19. Sun, Y.; Apostolaki, M.; Birge-Lee, H.; Vanbever, L.; Rexford, J.; Chiang, M.; Mittal, P. Securing internet applications from routing attacks. *Commun. ACM* **2021**, *64*, 86–96. [CrossRef]
20. Mehta, A.; Sandhu, J.K.; Pundir, M.; Kaur, R.; Sapra, L. Sinkhole Attack Detection in Wireless Sensor Networks. In *Proceedings of Data Analytics and Management*; Gupta, D., Polkowski, Z., Khanna, A., Bhattacharyya, S., Castillo, O., Eds.; Lecture Notes on Data Engineering and Communications Technologies; Springer: Singapore, 2022; Volume 91. [CrossRef]
21. Muzammal, S.M.; Murugesan, R.K.; Jhanjhi, N.Z.; Humayun, M.; Ibrahim, A.O.; Abdelmaboud, A. A Trust-Based Model for Secure Routing against RPL Attacks in Internet of Things. *Sensors* **2022**, *22*, 7052. [CrossRef] [PubMed]
22. Trofimova, Y.; Tvrdík, P. Enhancing Reactive Ad Hoc Routing Protocols with Trust. *Future Internet* **2022**, *14*, 28. [CrossRef]
23. Pathan, M.S.; Zhu, N.; He, J.; Zardari, Z.A.; Memon, M.Q.; Hussain, M.I. An Efficient Trust-Based Scheme for Secure and Quality of Service Routing in MANETs. *Future Internet* **2018**, *10*, 16. [CrossRef]
24. Hussain, M.Z.; Hanapi, Z.M. Efficient Secure Routing Mechanisms for the Low-Powered IoT Network: A Literature Review. *Electronics* **2023**, *12*, 482. [CrossRef]
25. Kenyeres, M.; Kenyeres, J. Distributed Flooding Algorithm for Sensor Fusion in Synchronous/Asynchronous Wireless Sensor Networks. In *Software Engineering Application in Informatics (CoMeSySo 2021)*; Silhavy, R., Silhavy, P., Prokopova, Z., Eds.; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2021; Volume 232. [CrossRef]
26. Kenyeres, M.; Kenyeres, J. Average Consensus over Mobile Wireless Sensor Networks: Weight Matrix Guaranteeing Convergence without Reconfiguration of Edge Weights. *Sensors* **2020**, *20*, 3677. [CrossRef]
27. Zhu, R.; Liu, L.; Song, H.; Ma, M. Multi-access edge computing enabled internet of things: Advances and novel applications. *Neural Comput. Appl.* **2020**, *32*, 15313–15316. [CrossRef]
28. Soundararajan, S.; Prabha, R.; Baskar, M.; Nagalakshmi, T.J. Region centric GL feature approximation based secure routing for improved QoS in manet. *Intell. Autom. Soft Comput.* **2023**, *36*, 267–280. [CrossRef]
29. Anbiah, A.; Sivalingam, K.M. Efficient failure recovery techniques for segment-routed networks. *Comput. Commun.* **2022**, *182*, 1–12. [CrossRef]
30. Verma, A.; Savita; Kumar, S. Routing Protocols in Delay Tolerant Networks: Comparative and Empirical Analysis. *Wirel. Pers. Commun.* **2021**, *118*, 551–574. [CrossRef]