Method of Routing Functions Virtualization in the Modern Networks

Kyryl Nikolchev, Yurii Ahafonov, Olena Starkova and Kostiantyn Herasymenko

Taras Shevchenko National University of Kyiv, Volodymyrska St, 60, Kyiv, 01601, Ukraine

Abstract

The article discusses the existing routing methods, their advantages and disadvantages. It was discovered that to solve the routing methods disadvantages it is necessary to use the concept of Network Functions Virtualization (ETSI ISG NFV). There were also showed the capabilities of future routing methods programming.

Keywords [1]

Routing method, static routing, dynamic routing, NFV, SDN

1. Introduction

Corporate computer networks unite a large number of local networks and computer systems with conflicting requirements for the quality of information exchange. As a result, building a corporate network based on a tree-like structure of connections between its subscribers is ineffective, since it leads to a low total load of the network channels.

During the routing process, routers consider several alternatives to get to one destination. These alternatives are the result of redundancy built into most network projects. Several paths are needed, so if one fails, other alternatives will become available.

The router also performs many other tasks:

- Connecting local networks to the global network.
- Network segmentation into separate broadcast domains, which increases the security, performance and controllability of such networks.
- Finding the best route for packet delivery over the network. Routing tables and dynamic routing protocols of different types are used to find the best route to the destination for different parameters.
- Network infrastructure. To improve network access, routers can create different servers, such as a DNS or DHCP server.
- Creating encrypted tunnels for data transmission. It is often needed to securely access a remote network by creating a VPN connection to the destination.
- Firewall and Intrusion Prevention System (IPS) etc.

Typically, the proposed classifications are based on several key characteristics and boil down to the following types of routing:

- Static or dynamic.
- Centralized, decentralized or hybrid.
- From source or step by step.
- Single-path or multi-path.
- Channel state and distance vector.
- Single-level or hierarchical.
- Intradomain and interdomain.

Information Technology and Implementation (IT&I-2021), December 01–03, 2021, Kyiv, Ukraine
EMAIL: nikolchev.kyryl@gmail.com (K. Nikolchev); yurii.ahafonov@gmail.com (Yu. Ahafonov);. elesta.tcs@gmail.com (O. Starkova); c.herasymenko@gmail.com (K. Herasymenko);
ORCID: 0000-0001-8985-2442 (O. Starkova); 0000-0002-9545-5272 (K. Herasymenko);


- Other.

It should be noted that these classifications only state the existing approach and do not show advanced routing methods.

2. Analysis of static routing

2.1. Main uses of static routing

Dynamic routing, of course, has several advantages over static routing. However, static routing is still used in networks today. In fact, networks typically use a combination of both static and dynamic routing.

Static routing has several main uses including:

- Providing ease of maintenance for the routing table on smaller networks that are not expected to grow to a large extent
- Routing to and from stub networks
- Using a single default route, used to represent the path to any network that no longer has a definite correspondence with another route in the routing table.

2.2. Advantages of static routing method

The advantages of static routing include minimal CPU processing, ease of understanding and configuration.

2.3. Disadvantages of static routing method

Among the disadvantages there are time-consuming setup and maintenance, possible configuration errors in large networks, need for administrator intervention to maintain changing routing information, poor scalability in growing networks, cumbersome maintenance, need to know the entire network for proper implementation [1-3].

3. Analysis of dynamic routing

To solve the problem of improving the quality of traffic transmission in the network, it is necessary to analyze the methods of dynamic routing, their functional features, the main advantages and disadvantages in order to determine possible negative phenomena in the network and methods of influencing them. When analyzing routing methods, it is obvious that single-path routing protocols, which use the classical algorithms of Dijkstra, Bellman-Ford, Shuurbale to find the shortest path, cannot be used as a way to balance the load (traffic) in the network, since their specificity is to transfer traffic only by the best route. In addition, in most cases, the path is chosen without taking into account the current load of other network resources. If the shortest path is already congested, then packets will still be sent this way, which will worsen the situation on the network. According to the method of routing, networks can use centralized, decentralized and hybrid routing.

3.1. Centralized routing

Centralized networks are built around a single centralized server/master node that processes all master data and stores user data and information that other users can access. From here, client nodes can be connected to the main server and send requests for data instead of executing them directly.

Centralized routing is implemented according to the principle of choosing the direction of movement for each packet by the network control center, and network nodes only perceive and implement the results of solving the routing problem. The advantage of this method is the ability to select nodes that are simple in structure, since they take minimal participation in the routing process. However, with an increase in the number of nodes, the complexity of organizing the centralized management of the data transmission network increases.

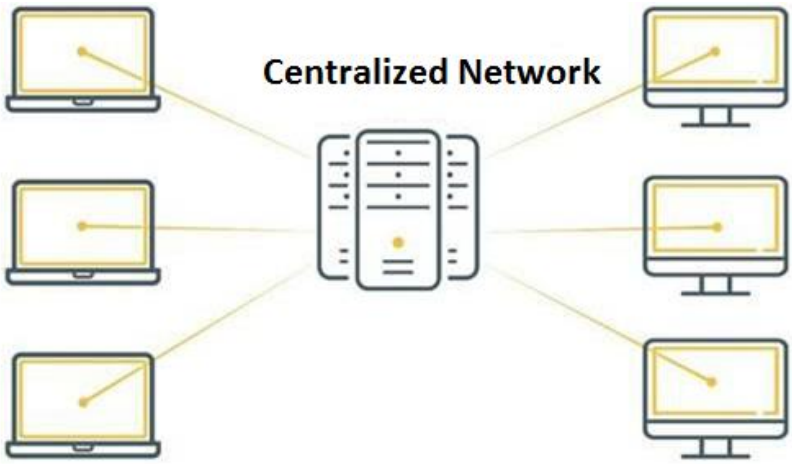

Figure 1: Centralized Network view

It is also easy to add and remove client nodes from the network by creating or removing connections between the client node and the main server. However, this does not increase the processing power of the network. Centralized networks tend to be the most cost-effective option for small systems and require fewer resources to set up and maintain. Also, when the network administrator needs to fix or update the network, only the central server needs to be updated. This reduces the time and overhead required to keep the network up to date.

Given the top-down nature of centralized networks, it is easier to standardize interactions between the primary server and client nodes. This can lead to a more consistent and streamlined end-user experience. In addition, since it is relatively easy to track and collect data online, a significant drawback of centralized control is the direct dependence of the quality of routing on the reliability of its control center, which tends to decrease with increasing complexity of the latter. In addition, the network control center must have operational information about the state of the network, since a node failure or its overload can lead to the loss of the entire network.

Since centralized networks have a single point of failure, if the primary server fails, the entire network is likely to go offline. Thus, client nodes will not be able to send, receive, or process user requests on their own. In addition, server maintenance may involve a temporary outage of the primary server, which is likely to result in service interruptions and, as a result, to inconvenience/decrease in reliability from the point of view of the user. Having a single point of failure also increases the chances of security breaches or disruptions due to cybersecurity threats such as DDOS attacks, as there is only one target that can be compromised. In addition, since there is only one central repository for user data, centralized networks will always carry inherent privacy risks. If the main server is damaged or out of service, its data can be irretrievably lost.

Centralized networks can be difficult to scale beyond a certain point, as the only way to do this is to add more storage, or processing power, to a central server. Moreover, if there are bursts of traffic on the network that exceed those that the network was designed to handle, information bottlenecks can arise, with users remote from the central server experiencing increased latency.

3.2. Decentralized routing

A decentralized network distributes information processing workloads across multiple devices instead of relying on a single central server. Each of these individual devices serves as a mini central unit that communicates independently with other nodes. As a result, even if one of the master nodes fails or is compromised, other servers can continue to provide users with access to data, and the entire network will continue to operate with limited or no disruption. Decentralized networks are made possible by recent technological advances that have provided computers and other devices with significant processing power and can be synchronized and used for distributed processing.

Distributed or decentralized routing is done through the distribution of network management functions among its nodes. Based on the stored control information, each node independently determines the direction of packet transmission. This increases the structural complexity of the nodes,

but the network is noted for a high level of availability, since the failure of any node does not affect the operation of the network as a whole.

Since decentralized networks do not have a single point of failure, they can continue to operate even if the master node is compromised or disabled. In addition, decentralized networks are easily scalable, since more devices can simply be added to the network to increase its processing power, and network maintenance usually does not require a complete network shutdown.

User requests are often faster when using a decentralized network because network administrators can create master nodes in regions with high user activity, as opposed to routing connections over large areas to a single centralized server. Decentralized networks provide a greater degree of user privacy because the information stored on the network is distributed across multiple locations, rather than a single location. This makes it difficult to monitor the flow of data on the network and eliminates the risk of attackers having only one target.

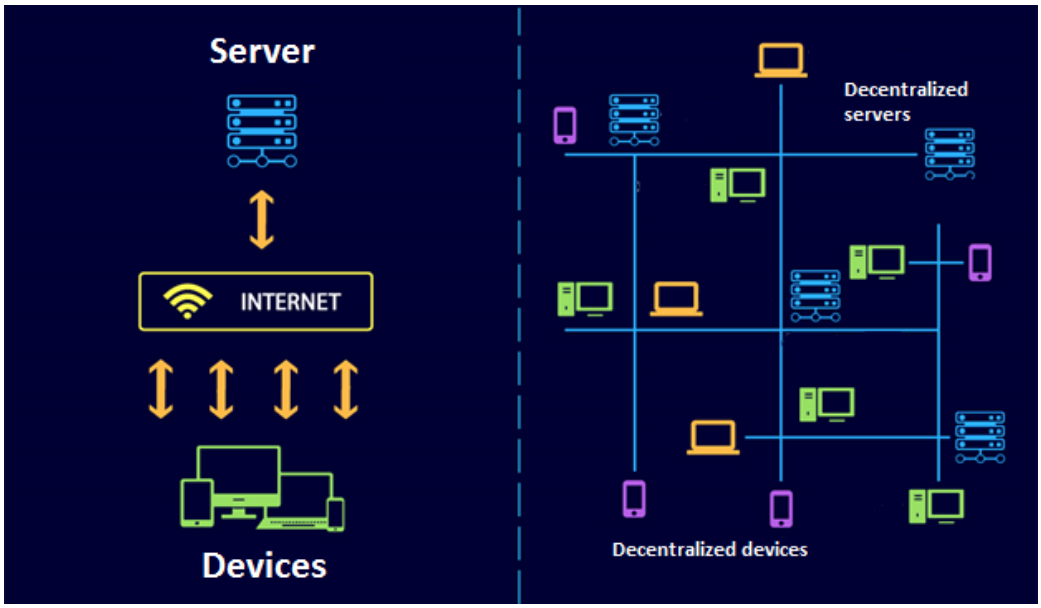

Figure 2: Centralized and decentralized networks comparison

However, decentralized routing has several disadvantages. Decentralized networks are more resilient than centralized ones. This usually makes maintaining these networks costlier and more time consuming. Since a decentralized network uses multiple devices to support the system, this places a commensurate burden on the organization's IT resources. As a result, decentralized systems are often not suitable for organizations that only require a small system because the cost-benefit ratio is not favorable under these conditions. Since the master nodes in a decentralized network operate independently and may not interact with each other, larger organizations can face coordination problems and find it difficult to manage and complete collective tasks. While this is a deliberate feature of decentralized networks, it means that not all business models and organizational structures will necessarily benefit from using a decentralized network.

3.3. Hybrid routing

Hybrid routing is characterized by the application of the principles of centralized and distributed routing (for example, hybrid adaptive routing). Adaptive routing involves adapting the routing algorithm to the real state of the network. The disadvantage of adaptive routing methods is the difficulty in predicting the state of the network.

4. Analysis of single-path and multi-path routing Methods

By the number of specific routes to one destination, routing protocols are divided into single-path and multi-path. Single-path protocols enter information about a single optimal route into the routing

tables. The obvious disadvantage is the uneven load of the network through the maximum load of the optimal route. Multi-path protocols are distinguished by the definition of several optimal paths. This makes it possible to parallelize the transmission of traffic and, as a consequence, to increase the reliability of data transmission and the efficiency of using communication channels. Despite the obvious advantages of multipath protocols today, modern networks use single-path protocols, the most famous of which are OSPF and EIGRP.

4.1. OSPF routing protocol

OSPF (Open Shortest Path First) is a widely used Interior Gateway Protocol (IGP) based on link-state technology and shortest path finding. This protocol carries out routing of packets, collecting information about the state of links from neighboring routers and, based on the information received, builds a network map. OSPF routers send many types of service messages, including hello messages, link status requests, updates, and database descriptions. The search for the shortest path is carried out according to Dijkstra's algorithm. OSPF uses a (cost) metric to select the best route, which is calculated based on the bandwidth of the link by default.

The advantage of transporting traffic when using OSPF is that network topology changes are processed very quickly. The main disadvantage of the OSPF protocol is that using Dijkstra's algorithm, one best route is determined, along which all traffic is directed. This can lead to congestion on the IP network and requires additional methods to be implemented.

4.2. EIGRP routing protocol

EIGRP (Enhanced Interior Gateway Routing Protocol), a distance vector dynamic routing protocol, has been optimized to reduce protocol instability after network topology changes, avoid route loop problems, and more efficiently and economically use router capacity and bandwidth. The composite metric, which is used to find the optimal path, is calculated based on throughput, load, latency, and reliability. This improves the quality of choosing the optimal route.

The main advantages of EIGRP are: low consumption of network resources in the absence of changes in the topology (only "hello" packets are transmitted) when changes occur, only information about the modifications that have occurred is transmitted over the network, which allows to reduce the load on the network and provides a short convergence time (in separate convergence is ensured almost instantaneously).

Along with the advantages of modern dynamic routing protocols, it should be noted that they all search for one best route with the minimum metric, that is, one-way, or balance routes in the network with the same metric, which causes the maximum use of the found best or alternative path and its overload. while other nodes (resources) of the network will not be involved in the process of traffic transmission. This approach does not make it possible to achieve a state of full equilibrium, a balanced distribution of the load between all possible alternative paths.

EIGRP provides mechanisms for implementing multipath routing, in particular through the unequal cost load balancing technique, but it is rarely used because it complicates the configuration process. In addition, dynamic link parameters such as reliability and utilization are not used by default when calculating metrics in EIGRP, since their use leads to constant changes in metrics and, as a result, route rebuilds. [1]

It is impractical to correct the situation by introducing changes to a specific protocol, since this problem is observed in all dynamic routing protocols, so a more effective solution would be to modify the routing process without making changes to a specific routing protocol. This option of influence will allow reducing the delay in traffic transmission and balancing the load on the network, universally for all dynamic routing protocols.

5. Other routing methods classifications

Single-level or hierarchical algorithms differ in how they interact with each other. In a peer-to-peer routing system, all routers are equal in relation to each other. In a hierarchical routing system, data

packets travel from lower-level routers to basic ones, which perform basic routing. Once the packets reach the general area of the destination, they are interleaved down the hierarchy to the destination host. In source routing systems, routers act simply as storage and forwarding devices for the packet, sending it to the next stop without any hesitation, they assume that the sender calculates and determines the entire route itself. Other algorithms assume that the sender's host knows nothing about routes. With this kind of algorithm, routers determine the route through the network based on their own calculations.

Intra-domain or cross-domain algorithms. Some routing algorithms only work within domains; others, both within and between domains. Link-state algorithms direct flows of routing information to all nodes in the network. Each router sends only that part of the information it knows that describes the state of its own channels, but to all routing nodes. Distance vectors require each router to forward all or part of its table, but only to neighbors [1].

6. SDN as a solution for routing methods problems

As we can see, there are no loss accounting methods among the existing routing algorithms. Dynamic routing reacts only to rough changes, and responds poorly to changes in channel congestion, delays are not taken into account, and the priority of the type of traffic is not taken into account even in EIGRP. Given a number of problems, and the limitations in the ability to solve these problems due to the impossibility of changing the standards, there is a need for tools to get around these limitations.

One of the most promising methods of traffic management in networks is the Software Defined Networking (SDN) model, which provides for the separation of traffic transmission functions and control functions, including control of both the traffic itself and the devices that transmit it. According to the SDN concept, all control logic is located in controllers that are able to monitor the operation of the entire network using special protocols (for example, OpenFlow), which operate on the concept of flows and can perform various actions with them (allow, deny, redirect, edit fields in packages, etc.). The advantages of a software-defined network are centralized management, simplification of network maintenance and modernization.

SDN can help because the goal of network management is to allow different devices (whether owned by a company, employees, or different manufacturers) to connect to networks and use their resources in a who-what-where-how-why-based manner. This requires consistent policy enforcement across all devices. Going forward, an administrator who changes policies will not have to spend hours making changes on each device separately, and these changes must be consistent across the enterprise. This is the role of SDN. They provide consistent, relatively fast network management by allowing changes across the entire network from a single management console.

It is also important that the network virtualization engine is built on the basis of free software, which allows network administrators to manage large data streams faster and more efficiently from a single console. Network functions virtualization (NFV) is an architectural framework created by the European Telecommunications Standards Institute (ETSI) that defines standards to decouple network functions from proprietary hardware-based appliances and have them run in software on standard x86 servers. Some of the benefits of NFV are similar to the benefits of server virtualization and cloud environments:

- Reduced capital expenditure (capex) and operational expenditure (opex) through reduced equipment costs and efficiencies in space, power, and cooling
- Faster time to market (TTM) because VMs and containers are easier to deploy than hardware
- Improved return on investment (ROI) from new services
- Ability to scale up/out and down/in capacity on demand (elasticity)
- Openness to the virtual appliance market and pure software networking vendors
- Opportunities to test and deploy new innovative services virtually and with lower risk.
  NFV Architecture Framework, that was developed by ETSI showed on Fig.3.

A simple router simulation was made using Python and sockets, simulating a very simple network with a single server and multiple clients [4-6]. The server shall be sending some data to the router, and the router will have functionality to decide which client to deliver the data to (Fig. 4-5).

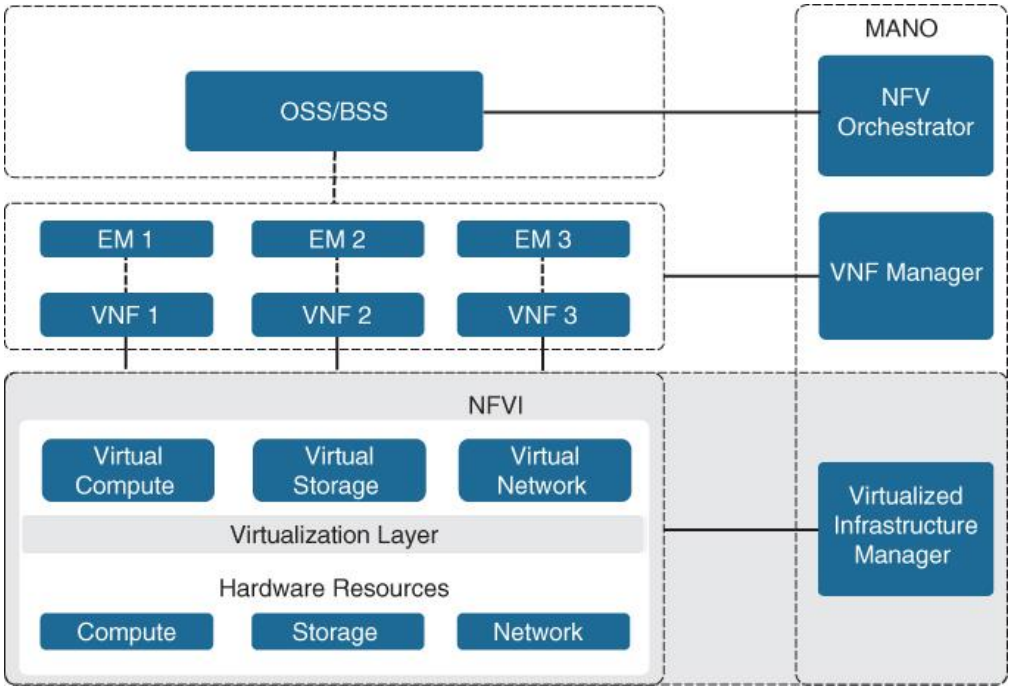

Figure 3: ETSI NFV Architectural Framework

```python
class Router:

    def __init__(self, hostname, interfaces):

        self.hostname = hostname

        self.interfaces = interfaces
        self.ip_list = [str(self.interfaces[i]["interface"].ip) for i in interfaces]

        self.__create_broadcast()

        self.data_packet = {"src_ip": "ip", "dst_ip": "ip", "data": "message"}
        self.broadcast_packet = {"src_ip": "ip", "data": "message"}

        self.connections = {}
        self.threads = {}

        self.routing_table = {ip_network('192.168.0.0/28')}

    def __create_broadcast(self):
        for interface in self.interfaces:
            broadcast_port = self.interfaces[interface]["br_port"]
            broadcast_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
            broadcast_socket.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)
            broadcast_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
            broadcast_socket.bind(('localhost', broadcast_port))
            self.interfaces[interface]["br_socket"] = broadcast_socket

    def connect_to_router(self, ip, l_port):

        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        sock.bind(('localhost', l_port))
        sock.listen(1)
        self.connections[l_port] = {"remote_ip": ip, "remote_port": l_port, "connection": sock}
        waiter = threading.Thread(target=self.wait_connection, args=(l_port,))
        waiter.start()
        self.threads[f"{l_port}_waiter"] = waiter

    def wait_connection(self, port):
        connection = self.connections[port]
        sock = connection["connection"]
        conn, addr = sock.accept()
```

Figure 4: A simple router simulation using Python. Sockets

Considering modern routing methods and algorithms problems and limitations in changing the existing standards the best decision is to use NFV [7-12]. Emulation of a router and its functions gives an opportunity to develop and modify our own standards, which gives us more flexibility while designing a corporate network [13-21].

```python
if __name__ == "__main__":
    sim = Simulation()

    r1_interfaces = {0: {"l_port": 9101, "br_port": 9910, "interface": IPv4Interface('10.1.1.1/24')},
                     1: {"l_port": 9102, "br_port": 9920, "interface": IPv4Interface('10.1.2.1/24')},
                     2: {"l_port": 9103, "br_port": 9930, "interface": IPv4Interface('10.1.3.1/24')},
                     3: {"l_port": 9104, "br_port": 9940, "interface": IPv4Interface('10.1.4.1/24')}}

    r2_interfaces = {0: {"l_port": 9201, "br_port": 9910, "interface": IPv4Interface('10.2.1.1/24')},
                     1: {"l_port": 9202, "br_port": 9920, "interface": IPv4Interface('10.2.2.1/24')},
                     2: {"l_port": 9203, "br_port": 9930, "interface": IPv4Interface('10.2.3.1/24')},
                     3: {"l_port": 9204, "br_port": 9940, "interface": IPv4Interface('10.2.4.1/24')}}

    r3_interfaces = {0: {"l_port": 9301, "br_port": 9910, "interface": IPv4Interface('10.3.1.1/24')},
                     1: {"l_port": 9302, "br_port": 9920, "interface": IPv4Interface('10.3.2.1/24')},
                     2: {"l_port": 9303, "br_port": 9930, "interface": IPv4Interface('10.3.3.1/24')},
                     3: {"l_port": 9304, "br_port": 9940, "interface": IPv4Interface('10.3.4.1/24')}}

    sim.add_router("router1", r1_interfaces, 9060)
    sim.add_router("router2", r2_interfaces, 9060)
    sim.add_router("router3", r3_interfaces, 9090)

    time.sleep(1)

    working_routers = {}
    for router in sim.routers:
        sim.routers[router].start()

    time.sleep(1)

    sim.routers["router1"].connect_to_router("10.0.2.1", 9101)
    sim.routers["router2"].accept_connection(9101, 9201, "10.0.1.1")
    sim.routers["router2"].connect_to_router("10.0.3.1", 9202)
    sim.routers["router3"].accept_connection(9202, 9301, "10.0.2.1")
    #
    sim.routers["router2"].message_to_connection("HELLO", 9201)
    time.sleep(1)
    sim.routers["router1"].message_to_connection("REPLY", 9101)
    time.sleep(1)
    sim.routers["router2"].message_to_connection("HELLO", 9202)
    time.sleep(1)
    sim.routers["router3"].message_to_connection("REPLY", 9301)
```

Figure 5: A simple router simulation using Python. Interfaces

```python
self.ip_list = []

for interface in self.interfaces:
    self.ip_list.append(str(interface.get_ip()))
    interface.hostname = self.hostname
    interface.routing_function = self.__parse_interface_data
```

Figure 6: A simple router simulation using Python. Router IP-addresses table

```python
# self.routing_table = {"network": {"gateway": "0.0.0.0", "interface": 0, "metric": 20}}
self.routing_table = {}
```

Figure 7: A simple router simulation using Python. Routing table

7. Conclusions

There is no simple test environment for creating new dynamic routing protocols. There are several projects for modeling a network on a regular computer, but they do not involve significant protocol changes and load the system with things that are not needed in routing testing, such as the channel layer level. Therefore, it was decided to create a simple system exclusively for testing dynamic routing

protocols. The system works on one computer and allows to quickly and easily create and modify various routing protocols with their subsequent testing, as well as in the future to implement the developed routing methods that are more efficient than existing ones. The reliability and validity of the results obtained by the author is based on the application of a systematic approach using mathematical models, methods of discrete mathematics. The practical applicability and significance of the conceptual and theoretical provisions developed by the author is confirmed by the fact that the developed methods are brought to practical implementation in the form of computer programs, which allowed to develop practical recommendations, namely for mathematical and software as part of information system.

8. References

[1] Network routing: algorithms, protocols, and architectures / Medhi D., Ramasamy K. San Francisco: Kaufmann Publishers is an imprint of Elsevier, 2007.-824 p.
[2] RFC 2328 OSPF Version 2, April 1998.
[3] Network Functions Virtualisation (NFV). [Electronic resource] - Access mode: https://www.etsi.org/technologies/nfv
[4] Use Containerlab to emulate open-source routers. [Electronic resource] - Access mode:
[5] https://www.brianlinkletter.com/2021/05/use-containerlab-to-emulate-open-source-routers/
[6] Creating a simple router simulation using Python and sockets. [Electronic resource] - Access mode: https://medium.com/swlh/creating-a-simple-router-simulation-using-python-and-sockets-d6017b441c09
[7] S. Russell, P. Norvig, Artificial Intelligence, A Modern Approach, Prentice Hall, 2003.
[8] J.F. Luger, Artificial Intelligence. Strategies and methods for solving complex problems, 2003.
[9] A. Teise, P. Gribomon, A logical approach to artificial intelligence: from classical logic to logical programming, 1998.
[10] R.Yager, D. Filev, Generation of Fuzzy Rules by Mountain Clustering, Journal of Intelligent & Fuzzy Systems (1994) pp. 209-219.
[11] N. Nilsson, Principles of Artificial Intelligence, 1985.
[12] K.Park, K.Lee, S.Park, H.Lee, Telecommunication node clustering with node compatibility and network survivability requirements, Management Science, vol. 46(3), 2000, pp.363-374.
[13] G. Luger, W. Stubblefield, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 2004.
[14] D. Poole, A. Mackworth, R. Goebel, Computational Intelligence: A Logical Approach, 1998.
[15] C. Wang, L. Ming, J. Zhao, D. Wang, “General Framework for Network Survivability Testing and Evaluation”, Journal of Networks, vol. 6, №6, 2011, pp. 831-841.
[16] Y. Kravchenko, V. Bondarenko, O. Trush, M. Tyshchenko, K. Herasymenko, O. Starkova, “Model of Information Protection system database of the mobile terminals information system on the territory of Ukraine (ISPMTU)”, IEEE International Scientific-Practical Conference Problems of Infocommunications Science and Technology, PIC S&T`2020 – Proceedings, pp. 785–790.
[17] D. Kovalchuk, Y. Kravchenko, O. Starkova, N. Tarasenko, K. Herasymenko, V. Riabtsev, “Development of recommendations for the implementation of virtualization concepts in modern networks”, IEEE International Scientific-Practical Conference Problems of Infocommunications Science and Technology, PIC S&T`2020 – Proceedings, pp. 797–802.
[18] G. Vlasyuk, Y. Kravchenko, O. Starkova, K. Herasymenko, A. Polianytsia, “Implementation of the Internet of things concept for remote power management”, 2nd International Conference on Advanced Information and Communication Technologies, AICT 2017 - Proceedings. pp. 26–30.
[19] Y. Kravchenko, O. Starkova, K. Herasymenko, A. Kharchenko, “Technology analysis for smart home implementation”, 4th International Scientific-Practical Conference Problems of Infocommunications Science and Technology, PIC S and T 2017 – Proceedings. pp. 579–584.
[20] Carl Hewitt. ORGs for Scalable, Robust, Privacy-Friendly Client Cloud Computing. IEEE Internet Computing. 12, N. 5, 96–99 (2008).
[21] Scarfone Karen. Guide to Intrusion Detection and Prevention Systems (IDPS) — 2007. — 127 p.