Tools for Analysis and Tracking of Deadlock-Free Routes in On-Board SpaceWire Networks

Valentin Olenev, Alexandr Karandashev, Kristina Alexeeva
Saint-Petersburg State University of Aerospace Instrumentation
Saint-Petersburg, Russia
{ valentin.olenev, aleksandr.karandashev, kristina.alexeeva}@guap.ru

Abstract—**This paper discusses the problem of tracking of deadlock-free routes by using various software tools. A brief overview of existing solutions providing this functionality is provided. A complete overview of the proposed software for building routes for given SpaceWire onboard networks is presented. The paper also discusses the mechanisms of calculating delays and choosing the best route. A brief overview of the methods for choosing the best route according to the provided criteria is given. A new method is proposed for choosing the best route, based on previously known methods and algorithms.**

I. Introduction

Nowadays there is a wide range of software that allows to track routes in networks. Programs such as AFDX Network Design Tool System, Application Programming Interface, and Microsoft Server Software - RRAS, etc. allow tracking of the routes for different networks. However, specialized networks need a field-specific software for routing assistance. Only such software will give ability to achieve reliability and accuracy in network design.

The design of aircraft and spacecraft on-board networks deserves special attention. The equipment responsible for people lives depends on reliable and accurate design. It is necessary to track routes in such networks taking into account data transmission protocols, equipment features and other. The product with such abilities is presented in current article.

This software complex is required when designing SpaceWire on-board networks. By means of this product it is possible to track deadlock-free routes for the implemented structure of the on-board network. However, lets first consider the existing program analogues.

II. Overview of Existing Solutions

A. Routing and Remote Access Service RRAS

Routing and Remote Access Service (RRAS) is the application programming interface and Microsoft server software [1]. RRAS allows to create applications that provide routing in IPv4 and IPv6 networks. In addition, communication between remote users and sites occurs through virtual private network (VPN) connections or remote access. Developers can also use RRAS to implement routing protocols.

Multi-protocol router is a computer running RRAS. It can track routes via IP, IPX, and AppleTalk simultaneously. RRAS includes two unidirectional protocols, Routing Information Protocol (RIP) and Shortest Path First (OSPF), and Internet Group Management Protocol (IGMP).

Remote Administration Program (RAS) provides remote connectivity by connecting via dial-up or through VPN remote access clients that use IP, IPX, AppleTalk or NetBEUI.

A separate router is usually used, so Windows Server also allows to configure the server as a routing device. The router function is to send packets from one network to the another. It does it by first determining the optimal route using routing algorithms. Routing algorithms determine the shortest "distance" (path with the lowest "cost") between the transmitter and the receiver of the packet. In addition, it supports routing tables that contain routing information to optimize information transmission.

Switches can also use route information change messages. It is used for communication with other routers to determine the best information routes sent from one network to another. In addition, it allows routers to compare and update their routing tables [1].

Routers and Windows computers configured as routers typically use static or dynamic routing algorithms. Demand-dial routing is also supported. All of these routing algorithms are basically the same purpose, although they have different mechanisms for sending information from transmitter to receiver.

This interface is not appropriate for our tasks because it can only be used on LANs and could not be used for SpaceWire.

B. AFDX Network Design Tool

Another software tool is presented in [2]. It allows designing of virtual channels in AFDX (Avionics Full-Duplex Switched Ethernet) network.

This tool defines limits on message latency, size and transmission periods. In addition, it takes into account the features of AFDX networks. Features such as the ability to use different buffer strategies on switch output ports or the ability to send multiple messages over a single virtual channel.

AFDX describes the control of message transmission in on-board networks based on the traditional Ethernet 802.3 standard.

This article presents the task of building multiple virtual channels and tracking routes for them in a data exchange network.

The system implemented in this article consists of two main modules: editing module and virtual channel build module.

Virtual channels and routes should have the following limitations:

1) The total bandwidth reserved for virtual channels passing through the physical data channel. It is should not be higher than its bandwidth.
2) The frequency frames transmitted for each virtual channel correspond rate at which messages are sent. This limitation occurs because all frames in the same message must enter the channel before the next message is sent.

However, virtual channels use a lot of memory. Each virtual channel should have a dedicated buffer in each switch. SpaceWire was designed to use as little memory as possible. This means the software discussed above is not appropriate.

C. Network Simulators (NS-3)

The NS-3 simulator is a discrete-event network simulator targeted primarily for research and educational use [3].

The general process of creating a simulation can be divided into several steps:

- Topology definition: To ease the creation of basic facilities and define their interrelationships, ns-3 has a system of containers and helpers that facilitates this process.
- Model development: Models are added to simulation (for example, UDP, IPv4, point-to-point devices and links, applications); most of the time this is done using helpers.
- Performance analysis: After the simulation is finished and data is available as a time-stamped event trace. This data can then be statistically analysed with tools like R to draw conclusions.
- Graphical Visualization: Raw or processed data collected in a simulation can be graphed using tools like Gnuplot, matplotlib or XGRAPH.

NS-3 used for to track the route. It use a Dijkstra Shortest Path First (SPF) algorithm in the topology for each node. In addition, it support simulation for TCP, UDP, ICMP, IPv4, multicast routing, P2P and CSMA protocols. However, this is not suitable for SpaceWire protocol.

Current software is not suitable for the SpaceWire on-board networks, so there is a need to have a more specialized tool.

III. SANDS SOFTWARE COMPLEX OVERVIEW

SANDS is a system for the design and simulation of SpaceWire onboard networks. It supports a full on-board network design and simulation flow that starts with network topology design and analysis and ends with getting of the results of network simulation [4].

Currently SANDS includes implementations of SpaceWire protocol and two transport layer protocols: RMAP [5] and STP-ISS [6].

The designed network is exported to an intermediate XML format for later use in other SANDS components for simulation, tracking of routs, or scheduling tables generation. The graphical interface can also display modeling results on a network structure, graphs and diagrams.

SANDS consists of four main components:

- Component for network physical structure design and setting of physical characteristics;
- Component for tracking of deadlock-free routes;
- Component for generation of STP-ISS transport protocol scheduling tables for data transfer with the Scheduling quality of service;
- Network simulation component, that processes all data that it got from the three previous components and the GUI [7].

Component №1 is responsible for the following tasks:

- Network topology design;
- Evaluation of physical characteristics of a designed network topology;
- Network transformation to achieve the required fault tolerance.

Network topology design process consists of creating a topology using the GUI, setting parameters and characteristics for nodes, switches, and channels. Once this step is completed, the network physical characteristics could be evaluated.

Component №2 is responsible for tracking of deadlock-free routes in the network [8].This is one of the most important tasks in networks design. It consists of finding a sequence of switches between transmitter and receiver and creating routing tables for these switches. SpaceWire technology provides the ability to create on-board networks with flexible architecture and high bandwidth. Configuring routing tables in switches allows to use static routing for a SpaceWire network. Moreover, the route tracking component gives ability to track the redundant routes.

According to the tracked routes, SANDS provide routing tables to configure the switches. In addition, this SANDS component provides a delay calculation for the worst-case of transmission of data and SpaceWire control codes.

There are only two transport-layer protocols for SpaceWire networks that provide time-division multiplexing mechanisms (STP-ISS, SpaceWire-D). Creating scheduling tables is a very difficult task, which is made more difficult with the growing network size and the intensive data exchange.

SANDS provide the functionality of scheduling tables' generation for the STP-ISS protocol (Scheduling quality of service). These tables take into account the created network topology and data transmission routes tracked in it.

SANDS component 4 provides simulation functionality. The simulation engine implemented using SystemC library.

Two network simulation modes with different levels of details are implemented:

- Bit-level – Simulation of the operation of the full SpaceWire stack, transport protocols and applications. This mode provides very detailed simulation that can take a long time.
- Packet-level - Simulation of top-level network operation only: Network, Transport, Application layers. This mode ignores some of the link layer details and provides quick simulation.

IV. MAIN ALGORITHMS

Component №2 is designed to track routes, create routing tables for switches and generate packet path addresses. It also provides the ability to change the parameters of physical devices and channels. It sets logical addresses for terminal nodes. In addition, it allows configuring parameters for transport layer protocols RMAP, STP-ISS.

In particular, component 2 allows:

1) Track main and alternative routes;
2) Calculate information flow latencies for the specified route configuration;
3) Set logical addresses for terminal nodes and regions;
4) Configure parameters for RMAP and STP-ISS transport layer protocols;
5) Change the settings of physical devices and channels.

In order to avoid deadlocks, the first step to routs tracking is to solve the problem of deadlock-free routing. The method of tracking deadlock-free routes in a network is based on building an acyclic graph of the network topology. The graph construction begins with a tree building, which sets the orientation of the network topology graph's ribs (channels). The tree is built in depth, starting from the root top.

Routing tables are created in accordance with routing up/down rules. The route should first pass through zero or more arcs in the straight direction ("up") and then zero or more arcs in the opposite direction ("down"). This avoids channel dependencies because the packet cannot cross the arc in the up direction after the arc is crossed in the down direction. This method of solving the deadlock-free routing task takes into account the wormhole routing used in the SpaceWire standard.

The provided functionality for building up/down routing deadlock-free routes is implemented in SANDS.

Two mechanisms are required to provide a user with information in a convenient and understandable way:

- Calculate the data latency for each route;
- Best data transmission route selection mechanism.

It is the purpose of current research. Let us look at the main mechanisms of component 2 in more detail.

V. UP/DOWN ROUTING

Up/Down routing is the deadlock-free routing algorithm for regular and irregular topologies. The SANDS uses routes tracking algorithm for the SpaceWire networks, that is differ from the classic Up/Down routing.

A classic algorithm builds a spanning tree for a given network by means of the BFS algorithm [9]. In our case, the spanning tree is built with the use of a Depth-first search (DFS) algorithm. It allows higher adaptability (less routing restrictions) and better traffic balance (fewer intersecting paths) to build routes. The DFS spanning tree is calculated from each node in the network. Then, for each tree the output channel is selected. Firstly, a channel to a node with a higher average topology distance to the other nodes is selected. The DFS is then marked as a tree with a lower average distance. When we get the required spanning tree, we should remove all cyclical dependencies. Then it is necessary to remove all cyclical channel dependencies. This will be done by splitting each loop. The other channel dependency determine the routing algorithm. The resulting routing algorithm follows the same up/down rule as the classic up/down routing algorithm.

VI. DELAYS CALCULATION MECHANISM

Modern on-board networks have complex irregular topologies. Tracking of the routes should be done in accordance with the designer's requirements. In addition, designer has to avoid packet blocking during the data transmission. Evaluation of correct routes on such topologies is a difficult task. In order to evaluate the route, it is necessary to calculate the data transmission delay for this route. Latencies in data transmission depend not only on the physical parameters of devices on the network. In addition, it also depend on how much data transmission will interfere with other flows passing through common parts of the route.

Latencies are calculated by getting the worst buffer states on routed switches. Each switch calculates the time the packet passes through the worst buffer state. The calculated latency are then summed up along the path of the entire route to get the resulting latency. This method allows to get the maximum delay when multiple packets in the same thread can be on the network.

A. Control Codes

The paper [10] provides an analytical model for calculating the latency for control codes in the SpaceWire network, where the control codes are: time codes, interrupts and interrupt acknowledgments. The authors of the article revised the equation presented in the work [11].

In paper [10], latencies depend on the length of the control code queue for the one (current) code. That's the number of higher priority codes of the same kind, and the length of the path between the transmitter and the receiver (network subscribers). Evaluation of the number of higher priority control codes is one of the advantages of this analytical model. With this analytical model, one can find minimum and maximum delivery delays and minimum, maximum, and worst control code propagation delays. An analytical model for calculating control code delays from an article [10] is represented by the following equation:

$$T\left(L_q\left(c\right),D\right)=L_q\left(c\right)*T_{cc}+\left(D-1\right)*T_{wtc}+D*T_{cc} \quad (1)$$

This model depends on the length of the queue — L_q, and the length of the path between the source and destination — D. Where T_{wtc} — the time of code passing through the router, not including the delay in waiting for the previous code to be transmitted. T_{cc} — the time when code is propagated over the channel.

B. Priorities of control codes

SpaceWire's on-board network could have three types of control codes, each of which has priority relative to each other. Time-codes have priority over acknowledgements and interrupts. Acknowledgements have a higher priority than interrupts.

The key difference between the information package and the control codes is that it has only one transmitter and only one receiver. It has less priority than time codes, interrupts and interrupt acknowledgements.

C. SpaceWire Data Transmission

The network switches processing of the packets that are being sent is done as follows: switch receives the first byte of the packet (header), then it routes the received packet to a particular output port. The other bytes of the incoming packet are then switched on-the-fly to the output port without packet buffering in the switch memory [12].

Thus, with the n bytes packet length the minimum delivery time of the packet Tmin can be evaluated as:

$$T\min=\frac{n*10+4}{F}+\sum_{i=1}^{S}th_i \quad (2)$$

where F is the minimum channel rate, measured in bit / s; th – is the time to process the packet header byte on the i- th of the routing switch; S – the number of network switches on the path from the transmitter to the destination node.

The fraction numerator in the equation (2) is ($n*10+4$) because thus data encoded in the SpaceWire standard. Bit stream consists of data and control characters. The character type is determined by the data/control flag value. The data symbol includes the parity bit, the data/control flag, and eight data bits [13].

Paper [12] also presents equations for calculating the maximum and worst latency for data packet transmission.

Thus, the equations above can calculate the data packet transmission latencies and control codes propagation latencies.

Now let us look at the best route selection mechanism.

VII. METHOD FOR THE ROUTE CHOICE

There are a number of different mechanisms for the best route choice. The study considered the following methods of choosing the best route:

- Method of estimation of routes by logical-probabilistic method;
- Comprehensive route selection method;
- BGP best path algorithm;
- Method based on routes derived from the genetic algorithm;
- Relief method.

The main requirements for these algorithms are:

- The dataset necessary to evaluate the route fits our task;
- The method takes into account message transfer latencies;
- The method takes into account the number of hops for message transmission;
- Method works with deadlock-free routing;

The task is also to select the best route from the list that was created using the Up/Down routing algorithm.

A. Method of Estimation of Routes Based on Logical-Probabilistic Method

A method designed to exchange information in a changeable network structure – between auto transport and cloud environments.

The MQTT protocol is used to exchange data. The probability of a message being sent is used as the route selection criterion. Depending on the terrain, the intensity of the exchanges and the volume of data transferred, the location of the relay, the probability of data transfer will be very different at a random time. Reliability changes over time, as network configuration does [14].

This method is not suitable for considered case, because messages should be guaranteed delivered to their recipients. This is an important condition for an on-board networks where reliability is one of the most important requirements.

Moreover, this method uses wireless data transfer technologies to transmit messages. For task is need wire data transfer.

In addition, a feature of this method is the high dynamics of subnet parameters change. In task the network does not change.

B. Comprehensive Route Selection Method

This method is designed to select a route taking into account the load of channels and noise during transmission in the communication channel. For this method, the most suitable route selection criterion is throughput.

This method is not suitable for the task in question. The method uses a wireless network. For task is need method which work with wire. In addition, the necessary parameters: signal, distorting signal, noise. They are absent in the given model of the onboard network SpaceWire.

C. BGP Best Path Algorithm

Border Gateway Protocol (BGP) is a routing protocol between standalone systems. A standalone system is a group of networks or a network with common routing policies and common administration. BGP is mainly used to exchange routing information for the Internet.

BGP is a reliable and scalable routing protocol. This can be proved by the fact that BGP is a routing protocol used on the Internet.

In this protocol, the first valid path is accepted as the best. Then selected path is compared to the next path in the list. This action do so until reach the end of the list of all [15].

This algorithm is not suitable for choosing a route in our case. It is designed for large networks with many parameters when choosing the best route. For the task is need smaller networks with lower number of parameters. But this algorithm can be an excellent example when constructing a new method.

D. Method Based on Routes Derived from the Genetic Algorithm

The following method could be used to select the best route from the list of given routes. First, it is necessary to build the best route by means of genetic algorithm [16].

However, this method is also not suitable for our task, as it does not take into account the latency in choosing the route. Moreover, a genetic algorithm may not find a really better solution within a reasonable time period. In addition, genetic algorithm does not take into account the deadlock-free routing required in given task.

E. Relief Method

Minimum path length is the main criterion for selecting a path in this method. It can be expressed by the number of transit channels. Moreover, other criteria may be used as main criterion.

In the communication network where dynamic management is done by the relief method, the following operations should be performed: creation of relief and its correction. Creation is performed at the moment of network start. In addition, it can be at the time of network development that is new switch nodes are put into operation. Corrections are made periodically in the course of network functions. Moreover, corrections are made in the event of damage or congestion on the network [17].

The specified method forms the relief from each network switch. As a result, each communication line has S heights. In the result, the communication line with the minimum height is the outgoing line, which select firstly. Lines with bigger heights are outgoing lines of communication, which select secondly, thirdly, etc.

It is necessary to select an outgoing low-weight line in each switch to find the shortest route from an arbitrary switch to switch A. The need to transmit information when creating relief between all switches is a disadvantage of this method.

An example of A-relief creation is shown in Fig. 1, as a simple circuit from switches and lines.

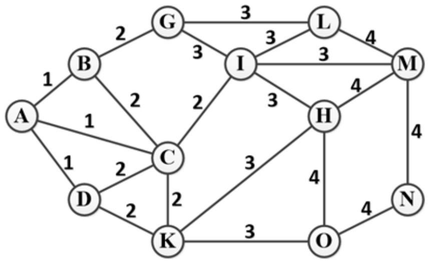

Fig. 1. Creation A – relief

This method the most appropriate for the given task. But it has disadvantage: it does not take into account the parameters of the message delay for each route. In addition, large networks will take a long time to calculate [18].

None of them can approach get task fully. Therefore, it is worth developing a special method of choosing a route from the given structure of the on-board SpaceWire network. It is necessary to take the best and most appropriate parts from each of the algorithms considered.

Five different methods of route assessment was considered. The results are presented in Table I.

None of them could be fully applied in our particular case. Therefore, it is worth developing a special method of choosing a route from the given structure of the on-board SpaceWire network. It is necessary to take the best and most appropriate parts from each of the algorithms considered.

TABLE I. OVERVIEWED METHODS COMPARISON

	Take into account delay	Take into account number of hops	Take into account deadlock-free routing	The given model has the necessary data	Supports wired data transmition
Logic-probabilistic method	+	-	-	-	-
Comprehensive method	-	-	-	-	-
BGP algorithm	-	+	-	-	+
Analysis of routes obtained by the genetic algorithm	-	+	-	-	+
Relief method	-	+	-	+	+

The following is the developed method of selecting the best route in the SpaceWire onboard network.

You must build the shortest route using the relief method. The number of hops should be used as a criterion for building the shortest route. The following analyses are possible:

1) 1 shortest route built.

- The shortest route should be compared to the Up/Down routing route list. If the route built is on the list, it is the best. Example is shown in Fig.2-a. In addition, due to the comparison with the list of routes from the Up/Down routing algorithm - it is possible to make sure that the built route will be deadlock-free.
- The shortest route should be compared to the Up/Down routing route list. If the built route is not listed, the

shortest route should be built by relief, excluding routes already considered. After receiving the new route, go to item 1.

2) 2 or more shortest routes built.

- One route is in the Up/Down routing algorithm list. The resulting route is the best route. Thanks to the comparison with the list of routes from the Up/Down routing algorithm, you can make sure that the route built will be deadlock-free.
- Both routes are in the Up/Down routing algorithm list. You must compare the latencies of the received routes in the Up/Down routing algorithm list. The route with the least latency will be the best. Example is shown in Fig.2-b. Thus, not only the number of hops and the deadlock-free routing, but also the time of transmission of data are included.
- None of the routes are in the Up/Down routing algorithm list. It is necessary to build the shortest route by method of relief, excluding routes already considered. After receiving the new route, go to item1. Example is shown in Fig.3-a.

3) All routes built were not found in the Up/Down routing route list. In this case, compare the routes from the Up/Down routing algorithm list. This will be the case only for deadlock-free routes. Find the route with the least data latency.

- Found 1 route with least latency. It is the best route for data transmission.
- Found 2 or more routes with equally low transmission latency. The route with the least number of hops in data transmission should be taken as the best. Thus, parmeters such as deadlock-free routing, latencies and the number of hops are taken into account. Example is shown in Fig.3-b.

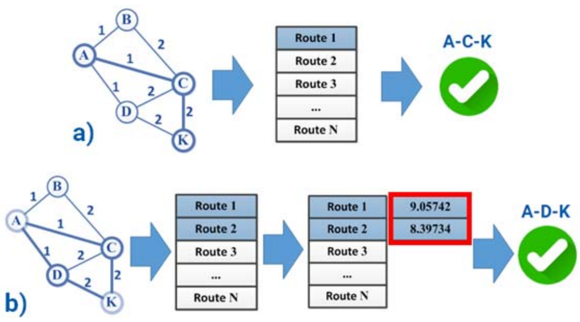

Fig. 2. First part of developed method

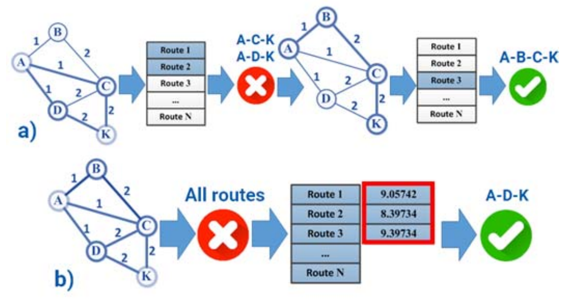

Fig. 3. Second part of developed method

However, the developed algorithm requires further update. It will be necessary to test the implementation on the given on-board SpaceWire network structure.

VIII. Conclusion

The paper gives a brief overview of various software tools that allow to build routes for networks. A detailed overview of the SANDS software complex is given. Similar software for routing are briefly described. The main mechanisms allowing to track, evaluate and select routes were presented.

Equations for calculating data transfer delays and control codes in the SpaceWire network have been reviewed. In future it is planned to clarify the calculation of latencies for other different network settings, for example, adaptive routing.

A generalized method of route selection is presented. We plan to implement the developed method in software for the testing and validation purposes.

Acknowledgment

The research leading to these results has received funding from the Ministry of Education and Science of the Russian Federation.

References

[1] Kabir M. H. et al. Detail comparison of network simulators //International Journal of Scientific & Engineering Research. – 2014. – Т. 5. – №. 10. – С. 203-218

[2] Makin D. S., Maklin J. Implementation, management and support of the network infrastructure Microsoft® Windows Server ™ 2003. - Russian ed., 2006.

[3] Vdovin P. M. Instrumental system for designing AFDX networks // Transactions of Moscow Institute of Physics and Technology. - 2015. - T. 7. - No. 2 (26).

[4] Olenev V.L., Lavrovskaya I.Y., Kurbanov L.I., Korobkov I.L., Sheynin Y.E. COMPUTER AIDED DESIGN SYSTEM FOR SPACECRAFT ONBOARD NETWORKS. Issues of radio electronics. 2018;(8):145-153. (In Russ.)

[5] ESA. Standard ECSS-E-ST-50-52C, SpaceWire — Remote memory access protocol. Noordwijk: Publications Division ESTEC, February 5, 2010.

[6] Y. Sheynin, V. Olenev, I. Lavrovskaya, I. Korobkov, D. Dymov "STP-ISS Transport Protocol for Spacecraft On-board Networks", Proceedings of 6th International SpaceWire Conference 2014 Program; Greece, Athens, 2014. pp. 26-31.

[7] A. Eganyan, E. Suvorova, Y. Sheynin, A. Khakhulin, I. Orlovsky, "DCNSimulator – Software Tool for SpaceWire Networks Simulation", Proceedings of International SpaceWire Conference 2013, 2013, pp. 216-221.

[8] Dally W. J., Seitz C. L. Deadlock-free message routing in multiprocessor interconnection networks. – 1988.

[9] J. C. Sancho, A. Robles, J. Duato, "A new methodology to compute deadlock-free routing tables for irregular networks", International Workshop on Communication, Architecture, and Applications for Network-Based Parallel Computing, Jan. 2000, pp. 45-60.

[10] Alexeeva K. I., Kurbanov L. I., Suvorova E. A. Calculation of Control Codes Delays in Onboard SpaceWire Networks //2019 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF). – IEEE, 2019. – С. 1-8.

[11] L.V. Koblyakova, S.V. Gorbachev "Tasks Of Hard Real Time Signals In Built-In Systems And Onboard Networks"

[12] "The upper limit of data transmission delays in SpaceWire networks" / A. S. Evdokimov, K. N. Rozhdestvensky / International Student Scientific Conference 70 / SUAI, St. Petersburg, 2018 (in Russian)

[13] L.V. Koblyakova, S.V. Gorbachev "Tasks Of Hard Real Time Signals In Built-In Systems And Onboard Networks"

[14] Glazunov V.V., Kurochkin M.A., Popov S.G. A method for evaluating message transmission routes in the telematic networks of

vehicles based on the logical-probabilistic method. Intellectual Technologies on Transport. 2015 No. (in Russian)
[15] Fabio Semperboni BGP best path selection URL: https://www.ciscozine.com/bgp-best-path-selection (дата обращения 04.12.2019)
[16] Zakharov V.V., Mugaysky A.V. Dynamic adaptation of the genetic algorithm for routing vehicles on large networks. Management of large systems. Issue 73. 2018 (in Russian)
[17] Kruk B.I., Popantonopulo V.N., Shuvalov V.P. Telecommunication systems and networks: Textbook. In 3 volumes. Volume 1. - Modern technologies in 2018, 620 pages (in Russian)
[18] I.P. Norenkov, V.A. Trudonoshin Telecommunication technologies and networks. 2nd edition, rev. and add. 2000 248 p. (In Russian)