INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING
DOI: 10.46300/9106.2022.16.43 Volume 16, 2022

Anti-jamming Wireless Sensor Network Model

Andrey Makashov, Andrew Makhorin, Maxim Terentiev
Department of Applied Informatics
Moscow Aviation Institute (National Research University)
Moscow, Russian Federation

Received: July 17, 2021. Revised: December 24, 2021. Accepted: January 11, 2022. Published: January 12, 2022.

Abstract—A wireless sensor network (WSN) of a tree-like topology is considered, which performs measurements and transmits their results to the consumer. Under the interference influence, the WSN nodes transmitters low power makes the transmitted information vulnerable, which leads to significant data loss. To reduce the data loss during transmission, a noise-immune WSN model is proposed. Such a WSN, having detected a stable connection absence between a pair of nodes, transfers the interaction between these nodes to a radio channel free from interference influence. For this, the model, in addition to forming a network and transferring application data, provides for checking the communication availability based on the keep-alive mechanism and restoring the network with a possible channel change.

A feature point of the proposed approach is the ability to restore network connectivity when exposed to interference of significant power and duration, which makes it impossible to exchange service messages on the channel selected for the interaction of nodes. To support the model, work algorithms and data structures have been developed, indicators have been formalized to assess an anti-jamming system work quality.

Keywords—Wireless sensor network, WSN, interference, data loss, channel change, reliability.

I. Introduction

Consider the IEEE 802.15.4 standard tree topology a wireless sensor network (WSN), which monitors a certain object. Its nodes perform measurements and send their results over a radio channel to the consumer. However, the radio channel is susceptible to interference, which the WSN nodes transmitters at low power can lead to significant data loss. To reduce data loss, some WSN nodes can change the used frequency channel and transmit their messages on a different channel, not affected by the interference influence The interference operation non-stationary mode forces to perform a channel change during the WSN operation. But the channel change must be performed in concert by at least nodes a pair - a transmitter and a receiver. There is coordinating no possibility' a change on the channel affected by the interference. It is also impossible to foresee a spare channel in advance, and it is also impossible to use it, since it is not known which channels will be affected by interference. In this regard, there is a need to develop a BSS model based on the self-organization principle, capable of independently changing the channel by some part of its nodes.

This possibility has been analyzed in some well-known works. The work [1] describes the channel switching mechanism using the spectrum transfer strategy. In the proposed protocol, the decision to change the channel is made by the coordinator, which does not correspond to the self-organization principle. [2] presents an algorithm for dynamic channel change, but it is applicable only to star and network topologies and takes into account only interference from the IEEE 802.11b standard devices. The algorithm proposed in [3] assumes that a node has several transmitters, which makes it inapplicable to nodes with one transmitter and receiver. The algorithm proposed in [4] uses the shared time slots principle, therefore, it requires nodes synchronization in time and a report transmission on channel occupancy to neighbouring nodes, which is not always possible in interference in certain types' conditions. In works [5] and [6], only the channels' loading by the network itself is taken into account, without taking into account external interference. Work [7] minimizes the switching number by channel nodes, however, the modelling did not take into account the significant external interference possibility. The article [8] describes a way to bypass the interference that potentially arises during the interaction of ZigBee and Wi-Fi networks. The method is based on the channel selection method based on the measurement of the channel load, for which the CCA data (clear channel assessment) is used. The article [9] proposes an adaptive channel change mechanism. The network is divided into groups of nodes, each of which consists of a parent node and one or more child nodes, while the parent initiates all channel change events. A common disadvantage of the above methods is that they all involve communication between nodes during the channel change process to inform neighbors or parents about the change. However, when exposed to significant interference power, a situation may arise in which access to the channel cannot be obtained, and any communication on the current channel cannot be performed.

Thus, in the known publications, there is no WSN model capable of changing the channel for the WSN nodes some part under the considered constraints and avoiding the external interference influence. This work is devoted to eliminating this gap.

The work further text is organized as follows. The first section is devoted to the model structure. The second section

presents a protocol stack, the network layer of which has been designed with channel change capabilities in mind. The third and fourth sections are devoted, respectively, to the network layer header structure and the application data transfer, also taking into account the possible channel change. The fifth and sixth sections are devoted to the formation and the network restoration. Channel change is presented in the seventh section. The solution quality evaluation to the problem is carried out using the indicators formulated in the eighth section.

II. Model Structure

Let's consider the proposed model general structure, connections between its various parts, input data and work results, see Fig. 1. The model basis is the wireless sensor network (WSN) nodes, model. The model defines the message format and communication algorithms between nodes. The nodes' main task is to collect, receive and transmit the environmental parameters measurements results. The nodes, in addition, interact with each other to form a network and restore logical connections if individual nodes fall under the interference influence. All nodes and noises are located in a certain area of space, which geometric parameters are determined by the applied problem.

WSN nodes and interference interact through a radio channel, determined by the data transmission frequency, as well as by the signal attenuation model depending on the frequency.

The model input data are represented by parameters in three groups: network parameters (area size, signal attenuation model, WSN nodes location and interference), WSN nodes parameters (operation algorithms, mode settings [10], etc.), interference parameters (operation schedule, radiation power, frequency channel).

The simulation results are measurements' a set collected by the network coordinator and integral indicators characterizing the WSN quality under the interference influence - the absolute and relative reliability values.

III. Protocol Stack

Network devices within the considered model framework interact with each other using the protocol stack, a four-level model. This model is an abbreviated OSI network model containing only the application (APL), network (NWK), link (MAC), and physical (PHY) layers. The model has been reduced from seven to four levels because the wireless network nodes have resources a limited supply and computing power. This leads to the fact that the model should be simplified as much as possible to improve the WSN performance. Data transfer between nodes is carried out in separate data packets, the data type "segments" is not used, so the transport layer is not needed in the model. The model does not provide for the initialization of communication sessions between nodes, since data transmission is carried out in separate packets, and the need absence for data presentation (except for the gateway, interaction is carried out only between network nodes, but not with external users) leads to the fact that for data processing, only applied level.

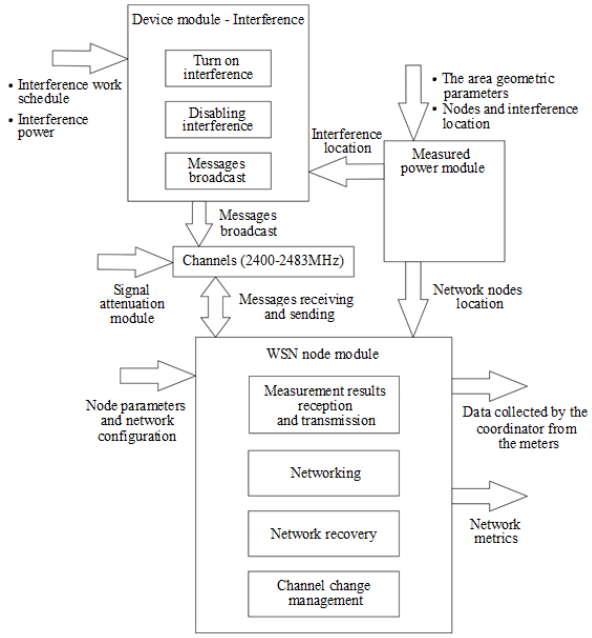

Figure 1. The noise-immune WSN model structure

Physical and link-layer models are under the IEEE 802.15.4 standard, including message formats, headers, frequency ranges, and more. Let's list the functions performed by the physical and link layers.

Physical layer functions: turning on and off the transmitter-receiver; the noise level estimation in the channel; the connection quality determination; the incoming signal power estimation; frequency selection for data transmission; packets reception and transmission through the physical medium; work with the physical layer headers.

Link layer functions: beacon messages control; access organization to the channel; guaranteed time intervals' management; checking the message format; confirmations delivery and acceptance; connecting to the network and disconnecting from it; mechanisms provision to ensure the information protection; work with the link-layer headers.

In the model under consideration, the following link layer functions are used: organizing access to the channel, checking the message format, delivering and receiving acknowledgements, and managing headers. The connecting and disconnecting functions from the network are not used within this model framework - these functions will be performed by the network layer.

Application and network layer models are not regulated by the IEEE 802.15.4 standard, they were developed separately and adapted to solve the problem under consideration. At the same time, as far as possible, the universality principle was observed, which makes it possible to customize the described model to solve problems in a wider range.

The network layer functions: logical addresses formation for nodes-candidates for connection to the network; network formation (consists in establishing logical links between nodes and assigning logical addresses to nodes); network restoration (consists in monitoring the parent and descendants activity to determine the communication lack fact due to interference or collisions and making a decision to disconnect from the network to find a new parent);

changing channels - selecting a channel number for communication and switching to other channels if necessary; processing the network layer headers (forming and adding the NWK header to the packet, reading and decrypting this header when receiving data); processing messages received from the data link layer, and transferring the message to the application layer, if necessary.

Application layer functions: the data sending scheduling, provided for by the solved application problem; data reception and processing provided for by the applied task to be solved; receiving data from NWK and transmitting data to it for further processing.

Next, the message headers, the model network and application layer processes are discussed in detail.

IV. Network Layer Header Structure

For a quick and correct analysis of the incoming data by the network layer to determine the message type and implement addressing, a network layer header is used, which is added to all messages transmitted over the network.

The network layer header consists of three fields: command, dest, source. The command field defines the message type and how it will be processed. The dest and source fields can be interpreted differently depending on the command type, the message nature, and the network operation mode. In general, these fields define the message recipient and sender logical addresses, respectively.

The PHY, MAC, and NWK headers have the following elements.

SHR (synchronization header) is a synchronization header that allows the receiving device to synchronize and block the bitstream. Consists of Preamble (preamble) and SFD (start-of-frame delimiter) fields. The first is responsible for data synchronization, the second signal the SHR completion and the data packet beginning.

PHR (PHYheader) - physical layer header containing the transmitted packet length. Its size is 1 byte, with 7 bits used directly to set the packet size, and 1 bit is reserved.

PSDU (PHYservicedataunit) is a variable-length field containing the transmitted packet data.

MHR (MACheader) - network layer header, consisting of several fields. FrameControl (control field) - a field that occupies 2 bytes and determines the frame type, sender and receiver addresses types, as well as other control flags. SequenceNumber (Sequence Number Field), also referred to as DSN (data sequence number), is a 1-byte field that identifies the frame sequence identifier.

Addressingfield - Addressing fields that define the network ID and short or extended MAC address of the sender and receiver.

AuxiliarySecurityHeader - a header connected in cases when it is required to protect a message with a certain setting in the control field. It is not used within this model, therefore its size will be 0 bytes.

MFR (MACfooter) - consists of one FCS (frame check sequence) field of 2 bytes and contains a value calculated based on the contents of the MHR and MACPayload fields, and thus is the checksum an analogue.

MACPayload is a variable-length field containing data transmitted by the data link layer.

Dest is a network layer header field containing, in general, the message recipient node logical address.

Dest is a network layer header field containing, in general, the message recipient node logical address.

A command is a header field of the network layer that contains information about the message type, and thus determines how it is processed by the network layer.

Payload is a variable-length field containing data, which composition depends on a specific application. This data is processed by the model application layer.

V. Transmitting Useful Data

By the useful data transfer, we mean measuring the environment physical parameters transmitting the results process by sensors over the network, or other interest information to the end-user in solving an applied problem framework.

The proposed subsystem main feature is that the transferring payload data process is separate from the service data exchange. Service data exchange includes data streams that are responsible for the formation, network restoration and channel change.

This feature allows you to transmit data with a frequency that is necessary when solving a specific applied problem. The interval for sending payload data by network nodes is t_{data}.

One of the main parameters, on which basis we will evaluate the network quality, will be the transmitted messages planned number (we denote it as k_{all}). If all nodes transmit messages at the same interval, then:

$$k_{all} = n\frac{t_{work}}{t_{data}},$$

where t_{work} is the total network operation time, n is the network nodes number.

The payload transfer process independence from the service data exchange processes allows each node or network segment to set its payload-sending interval (we denote it as t^{j}_{data}, where j is the network node number). Then:

$$k_{all} = \sum_{j=1}^{n}\frac{t_{work}}{t^{j}_{data}}.$$

The network layer header fields are used as follows: the command field uses a single DATA value to signal that a payload packet is being sent. The source field is used to store and transmit the node logical address that sent the packet, the dest field is used to store and transmit the node logical address to which the message is sent.

The maximum amount of payload L_{max} that the system can transmit is determined based on the IEEE 802.15.4 standard, which defines the PHY and MAC headers sizes, and the data maximum amount transmitted by the physical layer, as well as the NWK header size determined by the proposed model ... Then:

$$L_{\max} = aMaxPHYPacketSize - \\ - L_{MHR}^{data} - L_{MFR} - L_{NMK},$$

where aMaxPHYPacketSize is a constant defined by the IEEE 802.15.4 standard, $L_{\text{MHR}}^{\text{data}}$ is the link-layer MHR header fields size for the case of transmission of useful messages over the network, L_{MFR} field size level.

Payload messages are addressed using short addresses (2 bytes each in the MAC header for the sender and recipient host), therefore, a payload message maximum size is in bytes:

$$L_{\max} = 104.$$

The payload transfer process is initiated by the application layer. After the interval t_{data}, the application layer forms a packet with data, after which the generated packet is transmitted to the same node network layer, and the application layer, in turn, schedules the next event to send payload data. The network layer forms its header, entering into it the current node address, the coordinator's address (unless otherwise required for a specific task), and the DATA value in the command field. Next, the next node address in the route is determined, after which the packet with the NWK header and the data of the sender and the next node in the route, as well as other transmission settings, are sent to the data link layer by the McpsDataRequest message.

Data reception is initiated by the data link layer by sending a McpsDataIndication message to the network layer. This message is processed, the network layer header is extracted and decrypted in it. If the command field contains the value DATA, the message is classified as a payload and processing continue, otherwise, algorithms are used to process other message types. If the data is not intended for this node (it is an intermediate link in the route), the next node address in the route is calculated (see section VI**Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**), after which the packet with the same network layer header is sent back to the data link layer using the McpsDataRequest message. If the data is intended for this node, the message is transmitted to the application layer and processed under the applied task being solved.

VI. Networking

Under a network formation, we mean the assignment, as interaction a result with neighbouring nodes, to a logical address' each network node that uniquely determines a given node place in the interaction structure. Therefore, for forming a network processor with a complete description, it is necessary to determine the interaction protocol between nodes when connecting and an algorithm for choosing the logical address value.

The network is formed on a node initiative that is already connected to the network and has its logical address, see Fig. 2. Nodes that are not connected to the network wait for a broadcast invitation from the connected node and then send a request to connect to the network to the node from which they received the invitation. In turn, a node connected to the network, having accepted such a request, transmits data for connection, adding the node that sent the request to its descendants' list. An unconnected node, after receiving data for connection, saves the data of the inviting node and from that moment is considered connected to the network.

The sending invitations start to connect to the network is initiated by the coordinator. Further, his immediate descendants, accepting the invitation, also send such messages, which leads to the fact that in the end, all nodes connected to the network send such invitations.

The time interval that determines the frequency of sending broadcast invitations to the network by the coordinator is t_{inv}, while

$$t_{inv} = t_{base} \pm t_{rand},$$

where t_{base} is a reference value, $t_{\text{rand}} = U(0, b)$ is a uniformly distributed random component that reduces the service messages' simultaneous sending probability by several nodes, b is the upper bound of a random component. The introduction of a random component is necessary to reduce the load on the network, which inevitably occurs when several network nodes are sending broadcast invitations to connect at the same time. After a node that is not connected to the network sends a connection request, it waits for the time t_{w}, while all-new invitations from other nodes are ignored. If no connection data has been received during this period, the host resets the information about the inviting host and waits for the invitation again.

Possible values of the command field for the networking subsystem:

- BC_INVITE - broadcast invitation to connect to the network;
- BC_INVITE_NO_CONNECT - a broadcast message from a node that cannot connect to other nodes as streams, is ignored by potential descendants;
- REQUEST - connection request;
- CONNECTION_DATA - message with data for connection;
- DISCONNECT - a message for the node descendants being disconnected from the network with a directive to also disconnect from the network.

The network shaping subsystem uses the dest and source fields to transmit its logical address to the nodes that receive messages and to transmit the assigned logical address to the connected node.

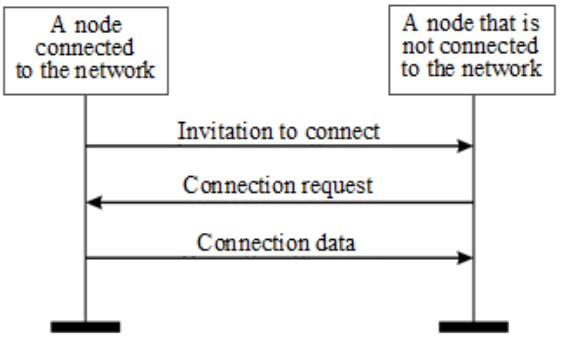

Figure 2. Messaging when connecting node

Consider the scheme for assigning logical addresses to

network nodes. The subsystem under consideration uses a logical address assignment scheme, in which the nodes each address depends on the parent's address. This approach advantages are that all routes are uniquely defined, and finding the next node address process in a route requires elementary computations to a minimum amount. Also, no auxiliary commands are required to transfer messages, which significantly reduces the traffic level on the network.

The nodes addresses are assigned as follows [11]: the coordinator (gateway) has the logical address "0". The address A_c node, which is node a descendant A, is determined by the formula:

$$A_c = A_m + k, k = 1\ldots,m,$$

where m is each node direct descendants' maximum number in the network; k is the descendant' number within the addresses parent's set.

A tree route generally consists of ascending and descending branches. In each of these cases, the next node address in the route can be determined by calculating the nodes parent addresses through which the route passes. Let's have a node with the address A. Then the address A_p of the parent of the node A is determined by the formula:

$$A_p = \left\lfloor \frac{A-1}{m} \right\rfloor,$$

where the operation ⌊⌋ means taking the integer part of a number.

VII. Network Recovery

The network recovery process refers to the connection detecting the loss process by the communication network individual nodes or segments with immediate parents or descendants, followed by disconnection from the network to find a new parent.

Network recovery is performed using the keep-alive mechanism, a variation [12]. The mechanism consists of periodically sending service messages necessary to check the node immediate descendants and the parent operability. Once every t_{SKA} node sends a broadcast message that can be received by neighbouring nodes. Each of the nodes that received such message increments the sender's keep-alive counter by one.

After the expiration of the time interval t_{KA} (the period for checking the keep-alive counters), a function is launched that checks the named counters for all neighbouring nodes. If the counter value equals zero, the node is considered non-working, if not, the node is considered operational, and the counter is reset to zero.

Thus, if a node does not receive a response from its parent within t_{KA}, it disconnects from the network and waits for new invitations. If the parent does not receive a response from the descendant within t_{KA}, it excludes it from its list of descendants and frees the address for new nodes.

To reduce the overhead messages traffic during network recovery, the same messages are used as during network formation. Depending on whether the node is connected to the network or not, the message will be processed in one way or another. Thus, $t_{SKA} = t_{inv}$, and henceforth we will use the notation t_{inv} to denote the interval for sending the combined service broadcast messages. The network restoration process also partially uses the same headers as in the network formation process. The list of possible values for the command field for the network recovery process is as follows:

- BC_INVITE - keep-alive broadcast message;
- BC_INVITE_NO_CONNECT - Broadcast keep-alive message from a node that cannot connect to itself as descendant nodes;
- REQUEST - response to the keep-alive message.

To assess the network restoration process impact on reliability, let us find the messages' estimated number that will be lost during individual nodes' reconnection due to the communication loss with the parent, leading to the transmitting messages impossibility to the coordinator from the network segment all nodes, the descendant to the node is disconnected. Let's find the time during which the node is unable to transmit useful data. Let's enter an auxiliary value t_0 - the minimum time required to complete one event processing and proceed to process the next. Let us consider further two cases: for the nodes no useful activity maximum and minimum periods (let us denote these values as t_{ua}^{max} and t_{ua}^{min}).

The maximum time is reached when the connection loss with the parent occurs immediately after receiving the last keep-alive message from this parent, and this, in turn, immediately after checking and clearing the keep-alive counter. In this case, the node waits for a period of t_{KA}, clears the keep-alive counter, and then has to wait for another interval t_{KA}, after which a disconnect decision is made. In the worst case, the node has to wait for another t_{inv} - t_0, if the invitation from the only available node occurred for some time immediately before resetting the counter and disconnecting the node.

Based on the above diagram:

$$t_{ua}^{max} = 2t_{KA} - 2t_0 + t_{inv} + t_{con},$$

where t_{con} is the time required to exchange messages when a node connects to the network, defined by the expression

$$t_{con} = \frac{L_{MHR}^{inv} + L_{MHR}^{red} + L_{MHR}^{cd} + 3\left(L_{MFR} + L_{NWK} + L_{SHR} + L_{PHR}\right)}{V},$$

where V is the data transmission rate.

When sending the BC_INVITE to command, the extended sender address and the short (broadcast) destination address are used, thus $L_{MHR}^{inv} = 17$ bytes. When sending REQUEST and CONNECTION_DATA commands, the sender short address and extended destination address are used, which gives bytes $L_{MHR}^{req} = 17$ and $L_{MHR}^{cd} = 17$ bytes. Considering that the data transfer rate in the standard under consideration is v = 250,000 bit/s, the values of the size of the physical layer header fields L_{SHR} and L_{PHR} is defined by the standard, and there is no payload in the service messages, we get t_{con} = 0.003168 s.

The values t_{ua}^{max} and t_{ua}^{min} are the time interval between the loss of connection with the parent (the node loses the ability

to transmit its data to the gateway) and the connection to the new parent (the node can again transmit its data).

The minimum time of inactivity will be achieved when the last keep-alive message arrives immediately before the keep-alive counter is reset, and the loss of communication with the parent is immediately after the reset. Then after one-period t_{KA} the node decides to disconnect. In the best case, the node accepts the new invitation immediately after disconnecting, so the interval t_{inv} is not taken into account in this case. From the above diagram:

$$t_{ua}^{\min} = t_{KA} + t_{con}.$$

Since the value t_0 is significantly less than the other values, it can be ignored when assessing the absence period. Thus:

$$t_{ua}^{\max} = 2t_{KA} + t_{inv} + t_{con}.$$

Based on the network parameter t_{tr}, the node transmits per unit of time the number of messages equal to $1/t_{tr}$. Thus, the number of messages lost during the absence of useful activity ($k_{ua}^{\max}$ and $k_{ua}^{\min}$) is determined by the following formulas (for the minimum and maximum values):

$$k_{ua}^{\max} = \frac{2t_{KA} + t_{inv} + t_{con}}{t_{tr}}, k_{ua}^{\min} = \frac{t_{KA} + t_{con}}{t_{tr}}.$$

For each node that has lost connection with the network, the inactivity time is in the interval:

$$t_{ua}^{\min} \langle t_{ua} \langle t_{ua}^{\max},$$

and the lost messages number is thus in the interval:

$$k_{ua}^{\min} \langle k_{ua} \langle k_{ua}^{\max}.$$

If the node is disconnected from the network has a child network segment, the messages lost number during the inactivity k_{ua}^{j} for the disconnected node descendant node at the level j is determined as follows:

$$k_{ua}^{j} = (j+1)k_{ua}.$$

The model provides for reducing this value possibility. For this, the following mechanism is used: when a node is disconnected from the network, it sends a broadcast message, which is received by the node immediate descendants. This message contains the DISCONNECT command, when received from the parent, the node is disconnected from the network. Thus, upon all shutdown commands successful transmission:

$$k_{ua}^{j} = \frac{t_{ua} + jt_{dc}}{t_{tr}},$$

where

$$t_{dc} = \frac{L_{MHR}^{dc} + L_{MFR} + L_{NVK} + L_{SHR} + L_{PHR}}{v}$$

there is the time required to transmit the shutdown command.

The DISCONNECT command uses the extended sender address and the short (broadcast) destination address, so $L_{MHR}^{dc} = 17$ bytes, so $t_{dc} = 0.001056$ s.

VIII. Change Channel

The channel change process means changing the frequency at which a node operates at the physical layer to bypass interference. This paper discusses algorithms for devices operating in the frequency range 2400–2483.5 MHz. Channels 11–26, located in this range, are available to the network nodes for transition. Thus, changing the channel by a node means changing the number of the channel on which the node operates, and correspondingly changing the frequency of the transmitter. In this case, a channel change in the model context can have two different types: a node constant transition to another channel, in which the node begins to receive messages on a new channel; a temporary transition to another channel to transmit a message, after which the node returns to the previous channel and receives messages on it.

Working with multiple channels is based on the following principles.

Each node has a predefined list of channels for sending connection invitations. Invitations are sent to each channel in this list in turn. There are two important parameters: n_{ch} - the available channels total number, $T_{S\backslash N}$ - the time between sending invitations to two adjacent channels from the list. These parameters correspond to the parameter T_{inv} for the single-channel method, while $T_{inv} \approx n_{ch}T_S$.

Each node "listens" to only one channel and is on it all the time, except for message transmission periods on other channels. If it is necessary to transmit a message on a channel other than the one that the node is listening to, then the node switches to the required channel to perform transmission, and then immediately returns.

This thesis is true for both disconnected and network-connected nodes. However, a node disconnected from the network changes the channel on which it is located if during the time $n_{ch}T_S$ it did not hear a single connection invitation. The channel changes to the next one from the same predefined list.

The list of channels for nodes to which the descendants' maximum possible number is connected changes. If it is not possible to connect to a site, there is no point in sending invitations to the entire available channel list. It is enough to poll the list of channels on which the immediate descendants and this node parent are listening to ensure the keep-alive mechanism works.

The keep-alive mechanism and address assignment work similarly to the single-channel case. As mentioned earlier, if a node does not receive keep-alive messages from its parent for some time T_{ka}, the node considers itself disconnected from the network. If a node during the same

period does not receive messages from its descendant, it excludes it from the list of descendants, at the same time sending it a disconnect command (which partially solves the problem if the connection is missing only in one direction). After disconnecting, the node starts listening to the next channel from the list and looking for a new parent to connect. Logical addresses are assigned in the same way as in the single-channel case.

This approach does not implement the scheme in which the node changes the channel without disconnecting from the network, for the following reasons: the channel on which the node listens for messages is known and used by both its descendants and the parent. If you change the channel, you must notify them all about it. But under the interference influence or even simply with a heavy load on the network, these messages may not reach the addressees, which will lead to network failures and a drop in reliability.

Node channel change is made under IEEE 802.15.4. For this, the standard structure PhyPibAttributes is used, which contains, among other things, the channel number on which the node operates. This structure is changed by sending a PlmeSetAttributeRequest, and the changing' the parameter fact is confirmed by receiving a PlmeGetAttributeConfirm message.

IX. Network Metrics

To assess the WSN work quality operating under the proposed model, two main indicators are used: relative and absolute reliability.

Relative reliability R_r is understood as the useful messages number ratio received by the network k_r to the useful messages total number sent by the network k_{tr}:

$$R_r = \frac{k_r}{k_{tr}}.$$

The absolute reliability R_a is understood as the useful messages number ratio received by the network to the planned sent messages number:

$$R_a = \frac{k_r}{k_{all}}.$$

The relative and absolute reliability values of the proposed interference-resistant WSN model outperform those of the conventional WSN. The specific numbers vary depending on the application in question for which the comparison is being made.

X. Conclusion

A noise-resistant WSN model is presented. Immunity to interference is provided due to the WSN segments' coordinated transition, affected by interference to other frequency channels. The model structure is presented, its components are considered. The WSN nodes protocols stack division into four levels is considered, their functions are described. Also, the main parameters affecting the WSN functioning and performance are determined. The network's work main indicators have been formalized, according to which its work quality is assessed.

WSN operating deployment on the proposed model basis will allow the creation of reliable sensor-control systems for the automation of production [13, 14] and maintenance of aviation [15] and space [16] equipment, will provide enterprises analytical systems [17] with reliable data on work processes, and will also serve as a data source for the various combined information products creation for end-users [18, 19]. In the listed areas of application, situations are possible when the network or its segment operates in close proximity to powerful sources of interference. The peculiarities of the network recovery algorithms make it possible to restore network connectivity even in this case due to the fact that the network can be rebuilt without the need for communication between nodes on the channel affected by the interference.

An increase in the WSN resistance to interference significantly increases the using the WSN possibilities, increases the data reliability received from the WSN, and, therefore, expands the tasks range in which the WSN, built on the proposed functioning model basis, can be used.

After the WSN functioning proposed model implementation, further work can occur in the WSN application researching new areas' direction, taking into account the new opportunities that have emerged. Also, the study can be the choice of the optimal input parameters of the system for various scenarios of the network functioning. Additionally, the proposed approach can be modified by the algorithm for compiling a list of channels available to nodes for transition based on the analysis of channel loading.

References

[1] P. M. Rodriguez, A. Lizeaga, M. Mendicute, I. Val, “Spectrum handoff strategy for cognitive radio-based MAC for real-time industrial wireless sensor and actuator networks”, Computer Networks, 152, p. 186-198, 2019.

[2] S. S. Wagh, A. More, P. R Kharote, “Performance Evaluation of IEEE 802.15.4 Protocol Under Coexistence of WiFi 802.11b”, Procedia Computer Science, 57, pp. 745-751, 2015.

[3] W. Wu, J. Luo, M. Yang, X. Li, “Energy-Efficient Channel Assignment with Switching Optimization in Multi-radio Wireless Networks”, IEEE International Conference on Systems, Man, and Cybernetics, 2015.

[4] C. T. Ngo, Q. L. Hoang, H. Oh, “BSSACH: A Big Slot Scheduling Algorithm with Channel Hopping for Dynamic Wireless Sensor Networks”, ADHOC-NOW 2017, LNCS, 10517, pp. 359-366, 2017.

[5] D. Nobayashi, Y. Fukuda, K. Tsukamoto, T. Ikenaga, “A Dynamic Channel Switching for ROD-SAN”, IEICE TRANS. INF. & SYST., VOL.E102-D, 5, 2019.

[6] S. Siddiqui, S. Ghani, A. A. Khan, “ADP-MAC: An Adaptive and Dynamic Polling based MAC Protocol for Wireless Sensor Networks”, IEEE Sensors Journal PP, 99, pp. 1, 2017.

[7] S. Yeoum, B. Kang, J. Lee, H. Choo, “Channel and Timeslot Co-Scheduling with Minimal Channel Switching for Data Aggregation in MWSNs”. Sensors 2017, 17, p. 1030, 2017.

[8] V. Kulkarni and S. K. Sahoo, “Load aware channel estimation and channel scheduling for 2.4GHZ

frequency band based wireless networks for smart grid applications", International Journal on Smart Sensing and Intelligent Systems, vol. 10, no. 4, 2017, pp. 879-902, doi: 10.21307/ijssis-2018-023.

[9] S. Yoon, R. Murawski, E. Ekici, S. Park and Z. H. Mir, "Adaptive Channel Hopping for Interference Robust Wireless Sensor Networks", 2010 IEEE International Conference on Communications, 2010, pp. 1-5, doi: 10.1109/ICC.2010.5502780.

[10] I. B. Ginzburg, S. N. Padalko, M. N. Terentiev, "Short Message Compression Scheme for Wireless Sensor Networks", 2020 Moscow Workshop on Electronic and Networking Technologies (MWENT), Moscow, Russia, pp. 1-5, 2020. DOI: 10.1109/MWENT47943.2020.9067371.

[11] S. N. Padalko, M. N. Terent'ev, "Self-organization in tree-like personal wireless networks in the presence of several gateways". Bulletin of the MSTU named after N.E. Bauman. Series Instrumentation, 1, pp. 75–85, 2017, DOI: 10.18698/0236-3933-2017-1-75-85.

[12] R. Braden, "Requirements for Internet hosts - communication layers", Internet Engineering Task Force, 1989.

[13] Sergey N. Padalko, Aleksandr A. Ermakov, Fatima M. Temmoeva, "Methodology for evaluating the effectiveness of integrated automation in the aerospace industry". Incas Bulletin, 12(Special Issue), pp. 135-140, 2020, DOI: 10.13111 / 2066-8201.2020.12.S.12.

[14] Yu. N. Kondrashov, A. A. Ermakov, "Optimization of planned solutions based on network models for large-scale problems". IOP Conference Series: "Materials Science and Engineering", vol. 862, article no. 052074. DOI: 10.1088 / 1757-899X / 862/5/052074.

[15] A. Stankevich, "A Model for the Operating Management of the Aircraft Maintenance Composition". In: R. Silhavy, P. Silhavy, Z. Prokopova, Ed. Software Engineering Perspectives in Intelligent Systems. CoMeSySo 2020. "Advances in Intelligent Systems and Computing", vol. 1294. (Springer, Cham, 2020). DOI: 10.1007/978-3-030-63322-6_89.

[16] Lubov Strogonova, Sergey Padalko, Yuri Vasin, Alexander Ermakov, "Technical and mathematical problems of microbiological protection of a manned space vehicle and stations". Incas Bulletin, 12(Special Issue), pp. 181-192, 2020, DOI: 10.13111 / 2066-8201.2020.12.P.17.

[17] Y. Kondrashov, O. Glushkova, D. Kobzev, "Planning and Approving Corporate Resource Development". In: R. Silhavy, P. Silhavy, Z. Prokopova, Ed. Software Engineering Perspectives in Intelligent Systems. CoMeSySo 2020. "Advances in Intelligent Systems and Computing", vol. 1294 (Springer, Cham, 2020). DOI: 10.1007/978-3-030-63322-6_86.

[18] I. Ginzburg, S. Padalko, M. Terentiev, "Combining Earth Remote Sensing and Land Wireless Sensor Networks Data in Smart Agriculture Information Products". In: R. Silhavy, P. Silhavy, Z. Prokopova, Ed. Software Engineering Perspectives in Intelligent Systems. CoMeSySo 2020. "Advances in Intelligent Systems and Computing", vol. 1294 (Springer, Cham, 2020). DOI: 10.1007/978-3-030-63322-6_88.

[19] I. Ginzburg, S. Padalko, "Use of a progressive web application for working with Earth remote sensing, topographic and cadastral data layers". IOP Conference Series: Materials Science and Engineering, 919(5), p. 052023, 2020, DOI: 10.1088/1757-899X/919/5/052023.