Simulation of a vessel sensor network of ZigBee standard

I I Buzenkov, Iu V Redkin, A A Tyufanova, E P Khaleeva

Admiral Ushakov Maritime State University, 93 Lenin Ave., Novorossiysk, 353924, Russian Federation
E-mail: igor.buzenkov@mail.ru

Abstract. The subject of the paper is a vessel sensor network designed to collect data from wireless sensors for various purposes. The peculiarity of this network application is the presence of a large number of premises, wireless communication between which is complicated. One of the possible approaches to solving this problem is to use coordinators connected by a switch into a local network, and a large number of routers that provide data relaying along the chain. Studies of the proposed vessel wireless network in the OMNeT ++ simulation modeling environment using Castalia framework are carried out in the paper. The process of modeling both the network itself and its individual components, and various modes of their operation is shown. The interaction of network nodes through intermediate routers is considered. Estimation of losses during packet delivery along a chain of routers, the time spent on processing packets in routers and delays in packet delivery to the network coordinator from the sending node is based on the compiled model. It is concluded that with the increasing complexity of the network structure (growth in the number of routers), there is a proportional rise of the packets delivery delay over the wireless network to the vessel local network. However, all transmitted packets are delivered to the addressee.

1. Introduction

Wireless networks are widely applied in various fields including shipping automation. The use of wireless sensor networks (WSN) is considered promising, these are distributed self-organizing networks that are resistant to the failure of individual network nodes. WPAN standards, and in particular the ZigBee standard based on IEEE group protocols are the most appropriate for use in these networks 802.15.4 [1].

ZigBee network on the vessel can be used to collect data from devices equipped with sensors: temperature, pressure, humidity, lighting, position and state of mechanisms, etc. Using wireless networks to compile information on the vessel has many advantages over traditional wired networks. The basic ones are the absence of cables, reduction of time and installation cost, low cost of network maintenance, ease of setting up and putting it into operation.

However, monitoring systems based on WSN must be reliable and have low data transmission latency, as well as resilient to external influences. Simulation modeling plays an essential role in the estimation of WSN parameters. Wireless network model of 802.15.4/ZigBee standard deployed on the vessel is considered in the paper. The OMNeT ++ simulator and the Castalia library, which model network behavior under realistic patterns of a wireless channel and media environment with fairly tangible behavior of network nodes, are used to estimate network parameters.
2. Research of WSN vessel structure of the ZigBee standard

ZigBee is an open wireless networking standard used to create data acquisition and control systems [2]. ZigBee technology allows creating self-healing and self-organizing networks that support automatic message relaying. It provides guaranteed packet delivery and protection of transmitted data at a low data rate.

Communication over the ZigBee network occurs using sequential packet relaying from the source node to the addressee node. Coordinators, routers, end devices are distinguished in the ZigBee network. The coordinator forms the network being its control center and trust center. It establishes a network security policy, installs settings when devices are connected to the network and provides communication with an external IP network, that is, it serves as a gateway for transmitting data collected from sensors. A router expands the coverage area of the network, performs dynamic routing bypassing obstacles, and restores routes in cases of network overload or device failure. Routers operate continuously, have a stationary power supply, and serve end devices including sleeping ones. The end device exchanges messages with routers, but does not translate or route packets. Sleeping devices that run on batteries in low power mode are recognized as a special type of device in ZigBee networks.

All network nodes have the ability to discover other nodes, identify each other and calculate the optimal path for data transfer, maximum communication speed, error rate and latency. The calculated values are transmitted to neighboring nodes, and the optimal transmission path is chosen due to the power of the received signals.

The generalized structure of vessel WSN based on the ZigBee network, designed to compile information from sensor detectors located on different ship decks, is shown in Figure 1. Here SR is the local network server (Server), SW – Switch, GT – Gateway, C – Coordinator, R – Router, S – Sensor. The coverage area of such a network inside the vessel premises is several tens of meters and depends on the number of used routers. To transfer information collected within the ZigBee network, other data transfer technologies are used outside of it. Since Ethernet is the main data transmission medium on the vessel, then an Ethernet gateway is required to connect the ZigBee network to the vessel local network. The application of gateways joined through a cable connection to the switch of the vessel network makes it possible to organize an extensive WSN consisting of several ZigBee networks that cover all decks and premises.

Figure 1 demonstrates that the proposed WSN provides data collection from sensor detectors distributed across various areas of the vessel. A feature of this network is the ability to quickly reconfigure the network depending on the state of the network nodes, the presence of obstacles in the path of signal propagation and interference. This improves the stability and reliability of the network.

![Figure 1. Structure of the vessel wireless network](image-url)
However, it is required to assess its efficiency, the promptness of data delivery to the vessel local network and operation reliability.

3. Development of WSN vessel model of ZigBee standard and its components

The basic tool for WSN analysis is simulation modeling without real equipment use which allows us to evaluate the projected network parameters at the stage of its development. There are lots of simulation modeling tools which are requested various demands. The OMNeT++ simulation modeling environment (simulator) is among them [3]. This simulator has an advanced graphical interface and provides flexible ability to change the parameters of the simulation model, and its functionality is not inferior to other simulation modeling tools.

The Castalia framework being the most requested library for simulating ZigBee networks, was used to simulate a vessel WSN in the OMNeT ++ environment [4,5]. The library model is based on veritable radio stations (using CC2420 microcircuits) for low power communication and supports a variety of modulation and transmission types. Functions of routing and environment access protocols, including IEEE802.15.4, are implemented in the library. The Castalia framework has a large number of configurable parameters and is designed to evaluate a variety of the network characteristics. The simulation outcomes of a wireless channel are very close to actual indicators, since they take into account various significant features of a tangible wireless channel.

A typical network, described by the Castalia library, assumes the use of nodes with the same set of parameters. It was used to simulate the WSN behavior using the OMNeT ++ simulator. The architecture of links of the model modules is displayed in Figure 2.

![Figure 2. Wireless network model](image)

Figure 2 shows that the Castalia library describes the sensor network as a set of nodes (Node), and besides each node is connected with sensors that control some physical process (Physical Process). Each node can be connected with one physical process. Sensor nodes communicate with each other by dint of a shared wireless channel (Wireless Channel).

The node model (Node) is presented as a composite object of the OMNeT ++ system, the node diagram is shown in Figure 3 (a). Components that implement application functions (Application), sensor functions for reading the process status (Sensor Manager), resource accounting functions - primarily electricity (Resource Manager), communication functions over a wireless network (Communication) and a node movement simulation unit (Mobility Manager) are constituents of the object.
The Communication component is also composite (Figure 3 (b)). The object components implement the functions of access control (MAC), access to the radio channel (Radio) and the routing module (Routing). All further components of the model are simple and implemented in C++.

4. WSN modeling in OMNeT ++ simulator

After creating the network for which the calculation is made, it is necessary to set the model parameters in the omnet.ini file, a fragment of the initialization file is shown in Figure 4.

```plaintext
[General]
include...Parameters/Castalia.ini sim-time-limit = 100s
include...Parameters/SensorDevice/Accelerometer.ini
SN.physicalProcessName = "CarsPhysicalProcess"
SN.physicalProcess[*].car_interarrival = 5
SN.node[*].Communication.Radio.RadioParametersFile = "/Radio/CC2420.txt"
SN.node[*].ApplicationName = "BridgeTest"
SN.node[0].Application.isSink = true
SN.node[*].Application.reportDestination = "SINK"
MyTest.node[0].Communication.MAC.isFFD = true
MyTest.node[0].Communication.MAC.isPANCoordinator = true
SN.node[*].Communication.RoutingProtocolName = "MultipathRingsRouting"

[Config Routing]
SN.field_x = 200
SN.field_y = 20
SN.deployment = "[1...34]->21x2"
SN.numNodes = 21
SN.physicalProcess[0].point1_x_coord = 0
SN.physicalProcess[0].point1_y_coord = 10
SN.physicalProcess[0].point2_x_coord = 200
SN.physicalProcess[0].point2_y_coord = 10
```

Figure 4. File sections of WSN model initialization

The model initialization file consists of sections; each defines a set of parameters for a separate experiment. The [General] section is common for all experiments. The used object is defined in the model settings when starting the simulation system. The RoutingProtocolName = "MultipathRingsRouting" parameter indicates the need to apply an object of the MultipathRingsRouting class when modeling. Simulation outcomes are recorded to a file.

The MultipathRingsRouting routing module was worked out in order to ensure that packets are transmitted along a chain of routers to a coordinator. The packet structure for routing is shown in Figure 5. Here the source is the sender's address; destination - recipient address; sequenceNumber;
multipathRingsRoutingPacketKmd is the routing packet type; sinkID - routing ID center; senderLevel is the level of the routing ring from which the packet is sent.

```
encapsulatedPacket = (MultipathRingsRoutingPacket) Multipath rings routing setup packet (cPacket)
controlInfo = NULL (cObject)
encapsulatedPacket = NULL (cPacket)
netMacInfoExchange (NetMacInfoExchange type)
    source = '1' [...] (string)
    destinations = 1' [...] (string)
    sequenceNumber = 0 (...) (unsigned int)
    multipathRingsRoutingPacketKmd = 2 (MPRINGS_TOPOLOGY_SETUP_PACKET) [...] (int)
sinkID = 0 (...) (int)
senderLevel = 1 [...] (int)
```

**Figure 5.** Routing packet structure

The packet transmission algorithm consists of two phases which are determined by the packet type. Setting up the links of nodes is done by installing their levels (rings formation) during the first phase. For this purpose, the coordinator sends a packet with the MPRINGS_TOPOLOGY_SETUP_PACKET type to the channel. A node, receiving such a packet, in case its level is not set, increases the obtained level by one, remembers it as the own and transmits the packet further. The rest of the nodes do the same. Thus, level rings are formed around the coordinator.

Data exchange takes place during the second phase. The node level is recorded when a message is sent from this node in a packet. If the node level is less than the packet level when receiving a packet with the NETWORK_LAYER_PACKET type, then the packet is transmitted further until it reaches the gateway. If the node level is greater or equal to the packet level, then such a packet is ignored (as a wrong direction). The algorithm serves to improve the efficiency of data delivery to the gateway and reduce the load on the network.

**5. WSN simulation outcomes in OMNeT++ simulator**

A number of experiments were carried out in order to determine the characteristics that designate the QoS parameter for assessing the performance of the developed WSN model (Quality of Service). The number of data packets that had been transferred in the first transmission cycle; the number of lost packets; Latency - processing time of an incoming packet at the node; Delay - the time required for the packet to leave the sender’s node and reach the coordinator were estimated as the research result.

The conducted experiments using the developed models confirmed the guaranteed delivery of packets from the sending node to the network coordinator.

Two parameters were monitored when evaluating Latency (stay duration of data packet at a node for its further forwarding to other nodes): moving time of a packet through the node from the application layer to the radio channel and the transit time of the node (the time taken to move from the radio channel to the radio channel).

Figure 6 (a) schematically demonstrates sending a packet from the application layer with the packet header analysis, changing the header if necessary, and finding a route to the destination node. The packet arrives at the radio layer in Figure 6 (b), it is analyzed and sent in the same way as in the case of sending it from the application layer. The result of modeling revealed that packet movement time from the application layer to the radio channel is 68.5 ms, and the time for the packet movement from the radio channel to radio channel is 75 ms.

Delay routing parameter shows the time of packet delivery to the destination node when it is forwarded through intermediate nodes (routers). The change in the packet transit time from the sending node to a destination node (coordinator) with an increase in the number of intermediate transit nodes (router nodes) was investigated during the simulation. Figure 7 shows an increase in the delivery time of a packet from a sending node to a destination node with an increase in the number of packet routing nodes between them.
Figure 6. Packet movement through the node: (a) - from application layer to radio channel; (b) - from radio channel to radio channel

Figure 7. Delay of packets on delivery due to packet routing

Figure 7 represents an increase in the number of intermediate routers due to a rise of packet delivery time, while the resulting dependence has a weakly expressed parabolic character (close to linear).

6. Conclusion

The paper considers a generalized structural diagram of the WSN network on a vessel taking into account its connection to the vessel local network for compiling data from sensor detectors located in various premises. The peculiarity of using WSN of the ZigBee standard on a vessel is the structure complexity of its connections, due to the presence of a large number of shielding partitions that impede wireless transmission of data between network nodes. One of the problem solutions is WSN structuring with the help of a group of coordinators located on various vessel decks, and the application of a large number of routers to relay data packets from sensor detectors. In the event of an obstacle, the solution allows restoring the violated paths automatically during data transmission and find an alternative path to the addressee without a significant time delay for restoring the network integrity.
The model describing the WSN operation on a vessel was developed on the ground of ZigBee network algorithm. Its research was conducted to evaluate the behavior of the WSN network under realistic transmission channel models in the OMNeT++ simulator and the Castalia library in order to determine the packet delays during their transmission over the wireless network. The experiments using the developed models confirmed the guaranteed packets delivery over the network, but indicated an increase in delivery time with a growth in the number of packet relaying. The resulting dependence of the delivery delay on the number of router nodes at its initial section has a character close to linear. It can be concluded that with the increasing complexity of the network structure (growth in the number of relays), there will be a proportional rise in the packets delivery delay over the WSN network to the vessel local network. Since the vessel WSN under consideration belongs to PAN networks characterized by a low data transfer rate, such a delivery delay in most cases cannot be considered critical. The most essential is the fact that according to the simulation outcomes, all transmitted packets will assuredly be delivered to the addressee.

References
[1] Alan Bensky A 2019 Short-range Wireless Communication (Cambridge: Elsevier).
[2] Vijayalakshmi S R, Muruganand S 2018 Wireless Sensor Networks: An Introduction (Mercury Learning and Information)
[3] Virdis A, Kirsche M 2019 Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem (Cham, Switzerland: Springer).
[4] Castalia Wireless Sensor Network simulator, available at: https://castalia.forge.nicta.com.au/index.php/en/
[5] Marco Z, Bhaskar K 2004 Analyzing the Transitional Region in Low Power Wireless Links Proceedings of the First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON), Santa Clara, CA.