Jakub MAJ[1], Krzysztof PIOTROWSKI[1], Emil MICHTA[2]
[1]IHP – Leibniz-Institut für innovative Mikroelektronik, Frankfurt (Oder), Germany
[2]Uniwersytet Zielonogórski, Instytut Metrologii Elektroniki i Informatyki

KANGAROO: MULTI-HOP PROTOCOL STACK FOR SMART CITY SENSOR NETWORKS

This work presents the description of the Kangaroo Multi-Hop protocol stack used in Wireless Sensor Network solutions. The article presents the requirements for the stack, its structure and functions of the stack, as well as the types of information stored on the nodes and transferred between the nodes. These mentioned aspects are described with a division into individual layers of the stack. The approach used in Smart City Sensor Networks is based on the Long Range physical layer.

KANGAROO: WIELOSKOKOWY STOS PROTOKOŁÓW DLA SIECI SENSOROWYCH SMART CITY

W pracy przedstawiono opis stosu protokołów Multi-Hop stosowanego w rozwiązaniach bezprzewodowych sieci sensorowych. W artykule przedstawiono wymagania stawiane stosowi, jego strukturę i funkcje, oraz rodzaje informacji przechowywanych w węzłach i przesyłanych pomiędzy nimi. Wymienione aspekty zostały opisane z podziałem na poszczególne warstwy stosu. Podejście stosowane w sieciach sensorowych Smart City bazuje na warstwie fizycznej dalekiego zasięgu.

1. INTRODUCTION

The motivation for the presented approach was to create a Multi-Hop Protocol Stack (Kangaroo) for Smart City systems [1], which are often deployed across entire cities or suburban areas and use an increasing number of data-transmitting devices. Growing areas of operation of Smart City systems mean that the use of Single-Hop protocols and protocols allowing data transmission over short distances would require the use of a large number of gateways. Such a solution is difficult to implement since gateways need a constant power supply and access to an external network (infrastructure) to send the received data to data storage centers. The solution to this problem is to create a protocol based on the physical layer that allows data transmission over long distances, while supporting multi-hop communication. Such solution is flexible and scalable, thus it supports deployments with different area coverage. This approach was used to design the protocol stack for the Smart City solution in the SmartRiver [2] project. In this deployment, it was decided to create higher layers of the protocol stack and to use the SimpleLink Long Range [3] physical layer encoding technique that trades data rate for sensitivity gains, which extends the communication range.

The following sections present the requirements and description of Kangaroo Multi-Hop Protocol Stack. The article concludes with a perspective on future developments.

2. REQUIREMENTS AND FEATURES FOR THE COMMUNICATION PROTOCOL STACK

The requirements and features established when developing the protocol stack for Smart City system are described in the order from the lowest to the highest layer of the protocol stack.

The first requirement was that the stack should be universal in terms of the physical layer. In this approach, the physical layer of SimpleLink LongRange and EasyLink HAL (Hardware Abstraction Layer) will be the basis of the communication protocol stack, but thanks to the use of a translation layer between the MAC and HAL layers, higher protocol layers, proposed in this work, will be able to cooperate with other physical layers as well.

The second requirement adopted when defining the communication stack is the use of a medium access protocol that guarantees reduced susceptibility to collisions. It was decided to use the

CSMA/CA [4] protocol (Carrier Sense Multiple Access with Collision Avoidance). The use of this protocol is dictated by the fact that the structure and operation of the entire stack are simplified because CSMA/CA is part of the radio drivers, and the upper layers are used for the CSMA/CA configuration.

The third requirement is the presence of the ACK (Acknowledgment) mechanism. This mechanism allows nodes to send an acknowledgment after successfully receiving the packet. Broadcast and ACK packets are exceptions here, and in the case of those packets, ACK will not be supported. This mechanism is fully supported by the MAC layer. As a result, higher protocol layers are not overloaded.

The fourth requirement is the support for multi-hop transmission that allows transmitting data with the use of other nodes being part of the Sensor Network. The use of multi-hop transmission allows data to be sent from more distant nodes or nodes located in places, where the radio signal is weak, without the need to install additional gateways.

To meet the requirements for multi-hop transmission, a mechanism for network structure (topology) building and routing must be established. The network structure building mechanism is periodically initiated by the designated gateway nodes by broadcasting the Discovery packet. After receiving such packet, the nodes in the vicinity decide if the address of the source node should be stored as a parent. This decision is based on the evaluation of the source node, which is done based on the hop-distance to the gateway and the RSSI of the received Discovery packet. After the decision is made and it is positive, the child node broadcasts its own Discovery packet with updated information about the hop-distance to the gateway. Each node broadcasts a Discovery packet only once for a single topology building procedure, except in situations where the node receives another Discovery packet with a better evaluation of the source node. In addition to participating in building the network structure, the Discovery packet will also be used to synchronize time between nodes.

After sending the Discovery packet, each node waits for a given configurable time for the Join packets. Join packet confirms that the node was chosen to be the parent by the child node that sends the Join packet and by that, these packets define the topology structure. If the wait time defined for the reception of Join packet expires without receiving any Join packet, then it means that the node is a leaf in the topology and it sends its Join packet to its best parent or all its parents (configurable option). If the parent node receives the Join packet, it saves the source address in its local children table, and sends its Join packet. Fig. 1 shows the mechanism for building the network structure with the packet sending order shown.

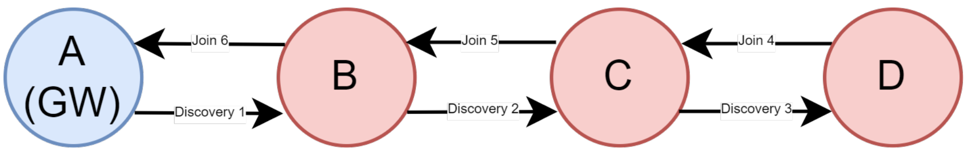

Fig. 1 Network structure building mechanism of Kangaroo Protocol Stack.
Rys. 1 Mechanizm budowania struktury sieci Stosu Protokołów Kangaroo.

Another configurable feature of the network structure building mechanism is the depth of the list of children sent in the Join packets. Thanks to that feature, the parent node can own a list of its grandchildren.

Besides the network structure building mechanism, both uplink and downlink routing is required to allow bidirectional multi-hop transmission of Data packets between the gateway nodes and other nodes in the network. Uplink routing, i.e. from nodes to the gateway, is based on the knowledge of the addresses of the parent nodes. A node sends the Data packet to its parent indicating that the destination address of the Data packet is the gateway address. Then, the parent does the same operation, until the Data packet reaches the gateway node. Downlink routing, i.e. from the gateway to the nodes, is based on the gateway's knowledge about all the paths created during the network structure building. The payload of the downlink packets stores the whole path from the gateway to the destination node. Each node relaying the packet, reads and deletes the first address from the payload and sets this address as the next-hop address. Then the node sends (the modified) packet to the next node until the packet reaches the destination node.

The fifth requirement is to store on the gateway node all the paths created during network structure building. The Information packets mechanism will help to meet this requirement. The Information packet is sent after each network structure building phase. This type of packet can store some data, but its main task is to collect the whole path from the last nodes on the paths to the gateway. Each node on the path, adds its address at the end of the packet payload and extends the length of the packet.

3. LONGRANGE MULTIHOP APPROACH

As mentioned in the previous section, the protocol stack is divided into layers. Fig. 2 shows the Kangaroo protocol stack along with the layers below and above the Kangaroo Protocol Stack.

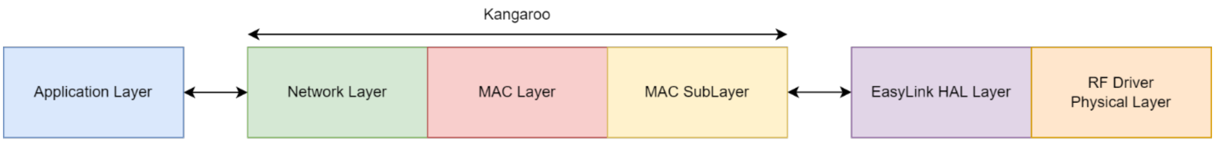

Fig. 2. Layers of Kangaroo Protocol Stack.
Rys. 2. Warstwy stosu protokołów Kangaroo.

Kangaroo, through the network layer, provides an API that allows higher layers to communicate with the protocol stack. The API allows the entire protocol stack to be initialized, initiates the process of sending data, and allows data from a received packet to be passed to higher layers. Initialization functions are designed to pre-configure the stack, allocate space for packets, and configure layers below Kangaroo. Functions enabling the initiation of the process of sending Data packets are to enable the transfer of data handled by the layers above the stack. The function responsible for passing data to the layers above the stack is called when the node is the target of the received packet. This data can be used by higher layers to change the node configuration. Between the MAC layer and lower layers, MAC SubLayer (translation layer) is placed. Due to this solution, the change of the HAL and/or the physical layer only forces the re-implementation of the translation layer, instead of changing the implementation of the entire stack. Besides network-related functions, Kangaroo uses system functions such as those provided by the clock and timer modules. These functions are used to control time-dependent operations.

Kangaroo uses many specific structures and variables to store configuration parameters and information. The configuration constants used by the network layer determine the maximum number of parents and children that a node can store in the array. The configuration constants common to Network and MAC layers specify the maximum size of a single packet and the maximum number of packets. The last configuration constant is used only by the MAC layer and is used to assign the maximum number of retransmissions in case of hop-by-hop transmission failure. Besides mentioned configuration parameters, Kangaroo allows configuring intervals between successive phases of (re)building the network structure and time to wait for the Join packets.

Kangaroo stores network information in four different structures. The first structure stores general network information or configuration, such as device address and broadcast address, minimum distance from gateway, number of children, and parents currently stored in the array. The second structure stores data about specific parents, such as the address of the parent, its distance to the gateway, the RSSI of the received packets from that parent and the link quality index calculated from the last two parameters. The third structure contains information about the children of the node. This covers the address of each child and the RSSI of the packets received from that child. The use of the fourth structure depends on the current protocol configuration because it stores the information about the grandchildren, and passing this information between nodes is optional. If this feature is enabled, then the structure stores the child's address, the number of its children (i.e. the number of grandchildren), and their addresses.

For the needs of the Kangaroo protocol, a packet structure was created that meets the assumed requirements. This structure is shown in Fig 3.

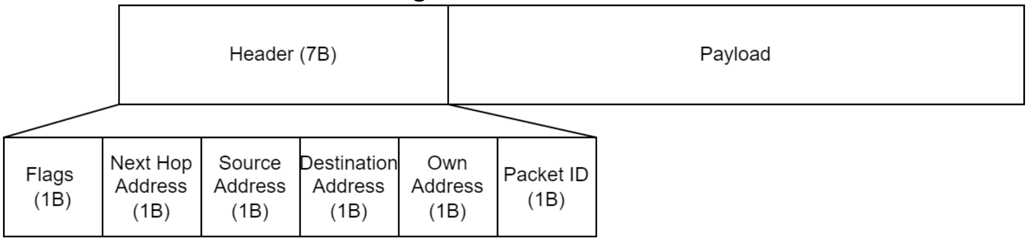

Fig. 3. Packet structure of Kangaroo protocol stack.
Rys. 3. Struktura pakietu stosu protokołów Kangaroo.

The structure of the packet used in Kangaroo consists of two main parts, i.e. the header and the payload. The header was structured in a way to contain all the fields needed for the functioning of the stack. The header contains the address fields, that is, the source and destination address, the address of the sending node, and the next-hop address. Moreover, it contains the control ID field. The last field in the header is a field that can contain up to 8 user-configurable true/false flags (i.e. packet type).

4. CONCLUSIONS

The Kangaroo protocol stack was designed for the use in a specific system and environment. The main assumptions of the protocol were to create a mechanism allowing the building of a network structure, enabling multi-hop communication both in uplink and downlink direction. Additionally, the protocol stack was designed to make its implementation, construction, and operation as simple as possible. One of the goals of the Kangaroo protocol stack was its versatility in terms of the physical layer used. However, in this particular scenario, where Kangaroo is used for data transmission in a Smart City system, it was decided to use the physical layer, which allows long-distance data transmission. This approach is dictated by the fact that the intermediary nodes can be deployed at longer distances from each other. As the next steps, the protocol will be evaluated in laboratory environment and finally in the real-life deployment in the SmartRiver project demonstrator.

ACKNOWLEDGMENTS

This work was supported by the European Regional Development Fund within the BB-PL INTERREG V A 2014-2020 Programme, “reducing barriers - using the common strengths”, project SmartRiver, grant number 85029892 and by the European Union ebalance-plus project under the H2020 grant number 864283. The funding institutions had no role in the design of the study, the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

BIBLIOGRAPHY

1. Mohanty S., Choppali U., Kougianos E.: Everything You Wanted to Know About Smart Cities, IEEE Consumer Electronics Magazine, 2016.
2. SmartRiver, O projekcie, http://www.smartriver.uz.zgora.pl/?page_id=11 (accessed 22.04.2022).
3. Hellan S.: CC13xx Long Range Modes Application Report, 2018.
4. Bertocco M., Gamba G., Sona A., Vitturi S.: Performance Measurements of CSMA/CA-Based Wireless Sensor Networks for Industrial Applications, 2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007, 2007.