IEEE 802.15.4 Thread mesh network – Data transmission in harsh environment

Wojciech Rzepecki
ABB Corporate Research Center
Starowiślna 13a
31-038 Kraków, Poland
wojciech.rzepecki@pl.abb.com

Łukasz Iwanecki
ABB Corporate Research Center
Starowiślna 13a
31-038 Kraków, Poland
lukasz.iwanecki@pl.abb.com

Piotr Ryba
ABB Corporate Research Center
Starowiślna 13a
31-038 Kraków, Poland
piotr.ryba@pl.abb.com

***Abstract*—the goal of the publication is to describe outcome form environmental tests of wireless Thread mesh network which was designed for indoor environment monitoring. The sensor is based on NRF52840 SoC (System on Chip) and allows to monitor parameters like: temperature, humidity, atmospheric pressure and light luminescence. For implementation of wireless communication OpenThread Thread protocol stack was used. All test were set in office building, where the harsh environmental condition, generated by hundreds of wire and wireless devices create perfect test place for verifying network reliability. The paper contains description of hardware and software solution, tests setup and outcomes concluded from collected data.**

I. 6LOWPAN

6LoWPAN is an acronym of "IPv6 over Low-Power Wireless Personal Area Networks". The 6LoWPAN concept provide Internet Protocol access for small, low power networks therefore the low power devices are capable of participating in Internet of Things communication and direct cloud access. From network layer point of view, the nodes in both IP and low power IEEE 802.15.4 networks are directly connected [1][2]. Every node is able to address any other node as well as external server and ordinary IPv6 network device. The example topology of network is shown in Fig. 1.

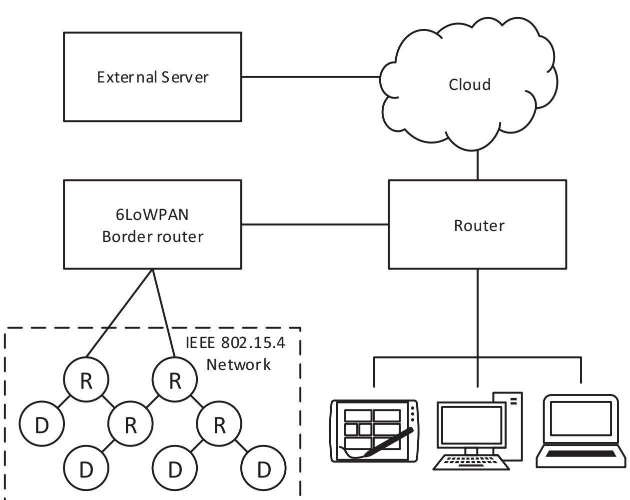

Fig. 1. 6LoWPAN example topology

The idea of transmission IPv6 messages in IEEE 802.15.4 networks impose us to consider several limitations caused by different capabilities of these two standards. First of all, the MTU (maximum transmission unit) in IPv6 is 1280 bytes [3]. The IPv6 messages are usually transmitted via Ethernet or Wi-Fi links which provide very high link rates. On the other hand, the standard MPU in IEEE 802.15.4 is limited to 127 bytes and throughput to 250kbps. Additionally, due to reduced header overhead and memory consumption, IEEE 802.15.4 standard supports two addresses: 16-bit short and 64-bit extended address. IPv6 usually runs at powerful stations with large amount of memory and computing power while IEEE 802.15.4 is expected to be power efficient and run in system with very limited resources.

In IEEE 802.15.4 standard, the maximum physical layer packet size is 127 octets. Maximum size of MAC layer header without security is 25 octets. It gives 102 bytes for payload. Further overhead is imposed by security layer. 21 octets for AES-128, 13 for AES-64, 9 for AES-32. It gives only 81 octets for payload [4][5]. On the other hand, in case of IPv6 the typical size of a header is 40 bytes. An IP packet usually encapsulate higher layers datagrams which requires additional bytes for their headers. For example UDP header is 8 bytes and TCP is 20 bytes. The overhead is far above IEEE 802.15.4 limits. Moreover, the MTU introduced in IPv6 is 1280 octets, so the additional fragmentation and adaptation layer must be provided.

To solve these challenges, 6LoWPAN standard was created. 6LoWPAN is specified in RCF4944 "Transmission of IPv6 Packets over IEEE 802.15.4 Networks". The standard defines frame format for transmission IPv6 packets and methods of formatting IPv6 addresses in low power IEEE 802.15.4 networks. There is an adaptation layer which allows IPv6 messages to be carried in efficient way in low power networks. Therefore, both internet and low power nodes, operate in the same subnetwork and share the same IPv6 address pool.

The main features of the 6LoWPAN are:

- IPv6 packet encapsulation
- IPv6 packet fragmentation – possibility of grouping one long IPv6 packet to smaller fragments that fit to IEEE 802.15.4 payload.

978-1-5386-7810-7/18/$31.00 ©2018 IEEE
DOI 10.1109/W-FiCloud.2018.00013

- IPv6 header compression – the compression method is specified in RCF6268. In the best case compression IPv6 and UDP headers can be shortened from 48 octets to 6 octets.
- Link layer packet forwarding – definition of mesh header

6LoWPAN introduces the following types of headers:

- Mesh header
- Fragmentation header
- Header compression header

II. THREAD NETWORK PROTOCOL

Thread is open standard of mesh network base at IPv6 and 6LoWPAN technology [6]. It introduces convenient way to build low power networks with direct access to the Internet. Thread uses IEEE 802.15.4 communication standard which is designed for low rate, low power WPANs (Wireless Area Networks). Moreover, Thread nodes use IPv6 connectivity that provide direct communication with external servers, clouds, user computer or mobile device. This feature allows to easy integration of Thread mesh network with other Internet network such as Local Area Networks. Basic topology consists of 3 major device types:

- Border Router (BR) – The gateway between Thread mesh and other IP based networks.
- Routers – With receiver/transmitter always on, linked with multiple connections between each other.
- End Devices – If configured as SED (Sleepy End Devices), can operate with low power consumption. Linked with only one parent router.

Example Thread mesh network topology is shown in **Fig. 2**.

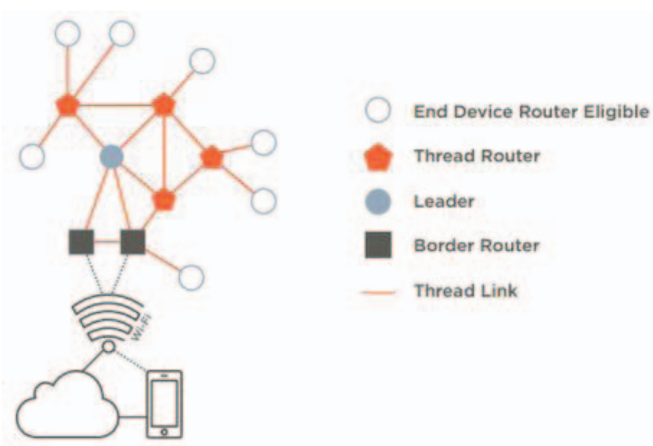

Fig. 2 Example Thread Mesh network topology [6]

The presented in this paper solution was designed to use it as a monitor of indoor environmental conditions. Sensors are programmed to making measurements one time in every one minute. The amount of transmitted data is not tremendous, up to one 128 kilobyte packet form single sensor per minute. The goal is to deliver all measured data to server. Data is buffered when the connection is broken and transmitted in time when the connection is stable or re-established. The stream of packets is inconsistent and non-real-time. The specification of application includes only long-term environment monitoring. It is not suitable for usage as feedback for control loops.

III. THREAD BORDER ROUTER

Thread border router is a gateway between Thread wireless network and LAN. In the project, Nordic Semi border router solution is used. The device is based on Raspberry Pi 3 model B microcomputer (RPi) and nRF52840 Preview Development Kit (PDK). The border router is controlled by the OpenWRT Linux distribution. The Nordic development board has IEEE 802.15.4 radio on-board and it is responsible for data transfer in the mesh network. The evaluation board is connected to RPi via Spinel protocol over USB. The firmware of the evaluation board is a full implementation of the Thread stack. 6LoWPAN packets from mesh side are converted to IPv6 packets and transfered to the LAN by software on the RPi [7]. The COAP (Constrained Application Protocol) server is running on the BR. COAP it is light protocol that operates over UDP and provides basic network reliability mechanisms. Due to harsh conditions of operation, proper ACK mechanism is crucial for this application.

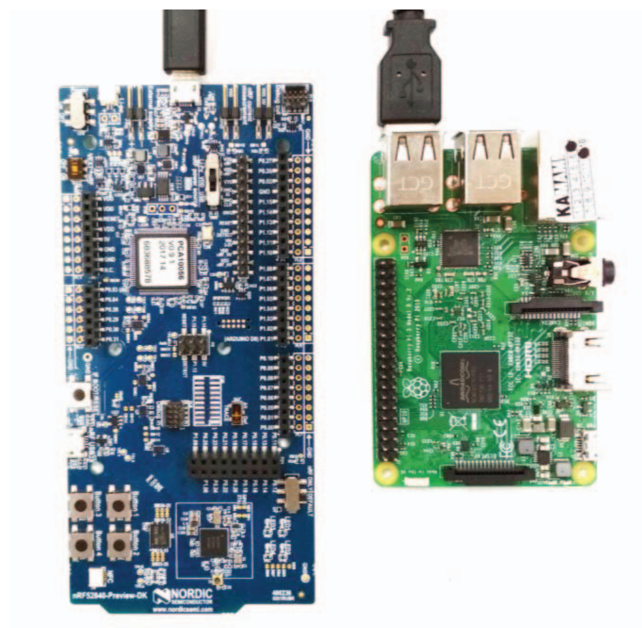

Fig. 3. Thread Border Router

COAP provides mechanism of acknowledgements as well as timeouts and retransmissions when a packet is lost. As a COAP server a Python library CoAPthon library is used [8].

For time synchronisation of the measurements, border router is capable of obtaining current time from NTP server and notify mesh sensors. Time synchronisation is done on demand by sensor device. The BR hardware setup is shown in **Fig. 3**.

IV. SENSOR DEVICE

In the project, were developed custom hardware for Thread sensor nodes. A heart of a node is Nordic Semiconductor nRF52840 SoC. The chip contains ARM Cortex-M4 core based microcontroller as well as integrated 802.15.4 radio transceiver in a single chip. It also provides additional IoT oriented features such as cryptography accelerator or Near-Field Communication (NFC). The SoC is dedicated for low power consumption and supports number of sleep modes. According to datasheet [9], current consumption is below 4µA during deep sleep mode and up to 10mA in radio transmission mode with +8dBi which is maximum output power. The nominal voltage of chip is 1.8V. SoC also contains LDO (Low Drop-Out) and DC/DC voltage converters. It can be powered from variety of sources from range of 1.8-5.5V.

Printed circuit board (PCB) is designed to be as small as the battery holder. It can be powered directly from coin battery CR2477N or via USB connector. PCB is made as 4-layer to provide the best electromagnetic compatibility and to simplify design process. Microstrip antenna design base at Texas Instruments [10] solution. Inverted-F PCB antenna limit the cost of each device and provides up to +3dBm gain for 2.45GHz frequency with omnidirectional gain characteristics. Sensor device visualisation is shown in **Fig. 4**.

Fig. 4. Sensor device board - visualisation

Board has embedded a few environmental sensors and it is capable of measuring the following parameters: ambient temperature, atmospheric pressure, relative humidity and light luminescence. The sensors are connected to the microcontroller via I^2C bus. Dataflow in application is shown in Fig. 5.

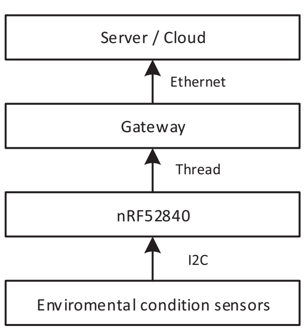

Fig. 5. Dataflow in system

The board may be powered from two sources: either from an ordinary 5V USB power adapter and from battery. It is possible to detect source of power supply in firmware and switch role of the node. When powered from a battery, the node acts as SED and operates in low power mode. Otherwise the node operates in Thread Router mode, never goes to sleep and it is used to build Thread network mesh skeleton. On-board analogue, power switches are designed to detect source of power supply and connect it to proper pins of the microcontroller. This mechanism is implemented due to provide greater energy consumption during sensor operation in router mode. Sensor role selection algorithm is shown in **Fig. 6**.

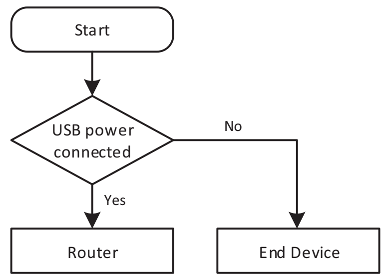

Fig. 6. Sensor role selection algorithm

V. SOFTWARE DESCRIPTION

Software solution delivered by Nordic Semiconductor consists of native drivers for nRF52840 and OpenThread implementation. In the project, OpenThread stack is used as a part of Nordic SDK for Thread. In the design, the Nordic SDK for Thread in version 0.11 is used.

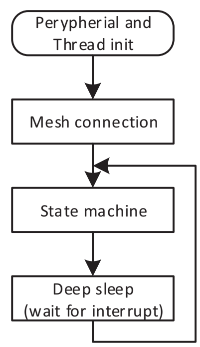

Fig. 7. Main program flow chart

The software is designed to work as a state machine. The diagram is presented in **Fig. 7**. After power up sequence and peripheral initialization, controller goes to idle state. It switches to deep sleep and stays in that state until an interrupt occurs.

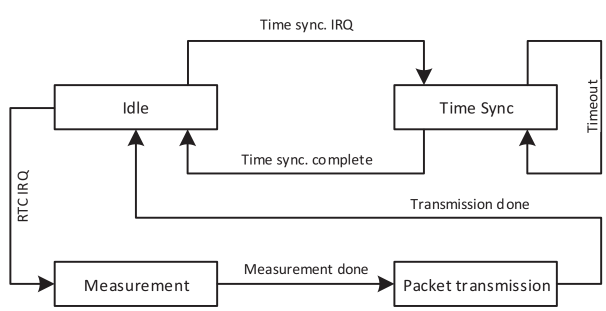

Fig. 8. State machine block diagram

The interrupt sources are RTC timer event or radio transceiver. Main program flow chart is shown in Fig. 8. Idle state is interrupted every one minute by RTC event. The state machine switches to measurement state where the measurements are triggered. When all measurements are done and frame forming process is finished, the system goes to transmission state. In this state the frame is sent as a COAP packet. Packet transmission state is terminated by acknowledgement or timeout event form COAP. In case of transmission failure, additional buffer is provided. All messages are buffered and removed from queue in case of receive proper ACK from COAP. Such an approach causes no loss in data stream. Data are complete in case of random packet loss or even temporary disconnection from rest of mesh network. Data transmitting algorithm is shown in **Fig. 9**.

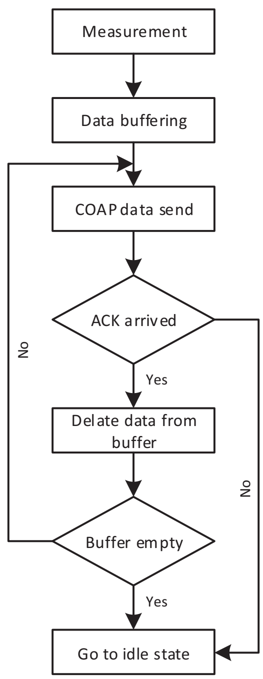

Fig. 9. Data transmission algorithm

Data is sent in JSON format to be usable by REST servers. Example of packet transmitted from sensor device:

{"A":"d46dc8f7dc6ac9fa","T":1511865080,"Stat":0, "THS":24.2,"Hum":25.3,"TPS":24.0,"Pres":985.4, "Lig":0.0,"Bat":-0.0,"Co2":0}

First field - “A” contains 64-bit device unique address. The next one have the following meaning: “T” - UNIX timestamp, “THS” – temperature from humidity sensor, “Hum” - relative humidity, “TPS” - temperature from pressure sensor, “Pres” - pressure, “Lig” - light density, “Bat” - battery voltage, “Co2” CO2 level in air (optional sensor).

The node involves RTC to keep current time and generate wake up events for a microcontroller. The node is capable of asking the BR for current time. Time synchronization is crucial in the application. Due to inconsistency of data flow, time of the measurement has to be provided in every packet. Nodes ask for time update in period of 12 hours. Keeping the nodes synchronised is important to avoid RTC time shift between sensor and server.

Microcontroller sends UDP packet time update request to BR. Then time server connects to specified NTP service and sends back current time as UNIX timestamp. Then the new time is set to RTC.

VI. Test Environment

The environment tests were placed is ABB Corporate Research Centre (CRC) office building. Research Centre office building contains several laboratories with equipment such as electric drives, power electronics, laboratory equipment, motors and radio transmitters which can generate strong electromagnetic noise. What is more, in the building is a lot of office equipment, such as PC, Wi-Fi devices like routers, smartphones, laptops or Bluetooth communication which operates on the same radio band as Thread network. Moreover, 2.4 GHz radio waves cannot easily propagate through solid materials such as brick walls and steel reinforced concrete ceilings. High suppression of 2.4 GHz band by walls and wall reinforcement significantly limits strength of the signal. It dramatically reduces the effective transition range and network reliability. Furthermore, this issues has large impact in wireless communication packet error rate.

ABB CRC building is 5 floors high with approximate 40m length and 15m depth. Big number of rooms, walls and reinforced concrete between floors reduce nodes’ effective range significantly. To provide reliable and redundant radio link, about 4-5 Thread routers are needed to be deployed at each floor. Moreover, deployment of the routers in rooms should be avoided. To achieve the lowest packet loss rate, the nodes should be placed in corridors where radio waves propagate directly through the open space. The routers build strong mesh network with redundant links inside the building. Battery powered SEDs are placed in random places.

In our example the Thread nodes send data to ABB CRC local cloud solution – Kibana 0. Kibana is a cloud inside ABB intranet where measurements from all local sensors are stored and processed. A screenshot of Kibana GUI and raw packets form the Thread network is shown in **Fig. 10**.

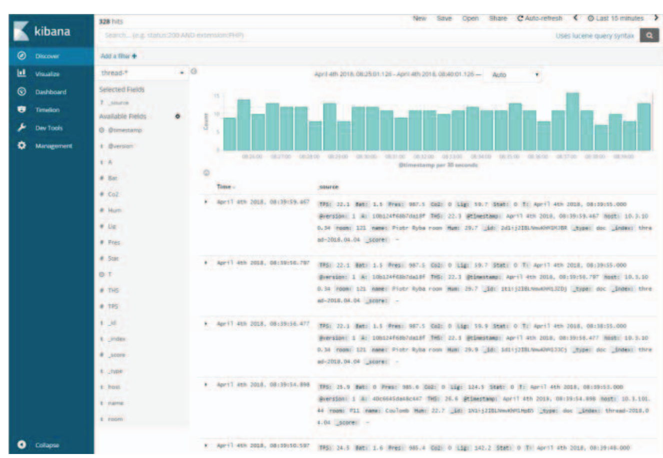

Fig. 10. Kibana data server interface

VII. TESTS SCENARIOS

There were two main tests made during development of Thread sensor device. First one was set multiple times to test short-term packet drop rate. Any major changes in communication algorithms were tested in this way. Purpose of this type of tests was to examine communication stability. At the beginning of development, simple UDP sockets were used to send data packets. UDP is datagram protocol that doesn't support acknowledgement and retransmission. More reliable communication was introduced by COAP transmission protocol. The goal of this test was comparison packet drop rate between trials and verification if any progress is made to obtain 0% packet drop

Second type of tests is long-term mesh stability observation with measurement of packet drop rate and consistency of data flow. The test was set as an output from whole case study project. Moreover, during placement of nodes, minimal number of sensors, to provide mesh stability, had been investigated. Connection between devices were monitored via Thread Topology Monitor - PC application delivered by NordicSemi. It uses PDK with serial communication to graphically show actual mesh network topology. Example topology is shown in **Fig. 11**.

Fig. 11. Thread Topology Monitor

VIII. TESTS OUTPUT

The **Table 1** shows comparison of pure UDP communication and COAP messaging. UDP without any ACK mechanisms does not provide 0% packet loss. Packets are often lost when strong disruption or mesh instability occurs. In second trial, with COAP and ring buffer, every sent packet arrived to server. The value above 100% is due to retransmissions when ACK from server is dropped. In this case, proper packet arrives but node not receive ACK so packet is send once again. It is worth to admit that in case of UDP packet loss or number of retransmission in COAP is greater when there is more nodes on the path of packet. In conclusion, packets are often lost in forwarding layer of routers.

Table 1 – Test results

Test no.	*Description*	*Packets received [%]*
1	UDP	88.15 %
2	COAP	101.05 %

Long-term test proved that all packets are arriving to server. Number of retransmissions, due to expectations, were strictly related with number of routers on packet way. Some of nodes, which were placed far away of the building, once in a while had temporary lack of communication. It can happen while mesh is rearranging. Due to usage of ring buffer, all messages were stored until successful transmission. Routers in number of 4-5 per floor were necessary to provide at least 3 links with other devices. Mesh stability rises when multiple connections are created.

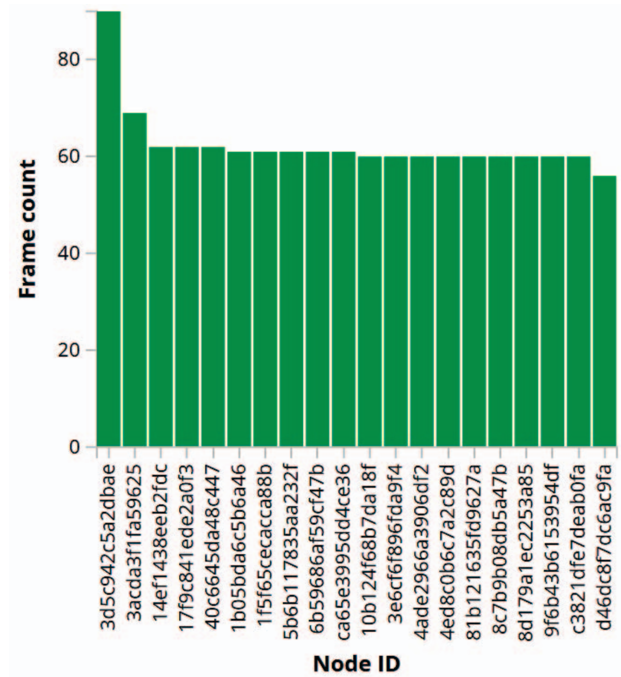

Fig. 12. Number of received packets per sensor in 1 hour

In **Fig. 12** numbers of received packets from one hour of tests are shown. For packets sent every 1 minute this value should be 60. Sensor with value below 60 has probably temporary lack of connection and it should send missing data when link is re-established. Value over 60 are due to retransmissions. In case of retransmission it is possible that a frame is sent to a receiver several times. This is probably caused by missing COAP acknowledge frames when the network is being rearranged. The most important is certainty that none frames are lost. All extra frames are easily detected and dismissed be a server. Thanks to the COAP protocol is certainty that all packets are received by a cloud.

IX. CONCLUSION

Case study presented in this paper proved that Thread mesh connectivity is good for low-power, low-cost and non-real-time measurements solutions. Weak point of this technology is operation in 2.4GHz radio bandwidth. This type of signals are easily distorted by many other radio devices in crowded environment. What's more, 2.4GHz band is attenuated through common construction materials such as bricks or reinforced concrete. Packet loss measured during tests was also correlated with number of devices on route of transmission. Used implementation of Thread is at early stage of development and future software updates of stack can increase reliability of tested protocol.

REFERENCES

[1] IEEE 802.15.4-2015, *"IEEE Standard for Low-Rate Wireless Networks"*, 2015

[2] N. Kushalnagar, G. Montenegro, RFC4919 – "*IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals"*, Aug 2007

[3] S. Deering, R. Hinden, RFC2460 – *"Internet Protocol, Version 6 (IPv6) Specification"*, Dec 1998

[4] G. Montenegro, N. Kushalnagar, RFC4944 – *"Transmission of IPv6 Packets over IEEE 802.15.4 Networks"*, Sep 2007

[5] J. Hui, P. Thubert, RFC6282 – *"Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks"*, Sep 2011

[6] Thread group, Thread specifiaction, https://www.threadgroup.org/ (accessed 01.04.2018)

[7] Nordic Semiconductor Infocenter, Thread border router, https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.threadsdk.v0.9.0%2Fthread_border_router.html (accessed 01.04.2018)

[8] G.Tanganelli, C. Vallati, E.Mingozzi, *"CoAPthon: Easy Development of CoAP-based IoT Applications with Python",* IEEE World Forum on Internet of Things (WF-IoT 2015)

[9] Nordic Smiconductor, NRF52840 Objective Product Specification, http://infocenter.nordicsemi.com/pdf/nRF52840_OPS_v0.5.pdf (accessed 01.04.2018)

[10] Texas Instruments, 2.4 GHz inverted-F antenna application report, http://www.ti.com/lit/an/swru120c/swru120c.pdf (acessed 01.04.2018)

[11] M. Bajer, *"Building an IoT Data Hub with Elasticsearch, Logstash and Kibana"* ,2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Prague, 2017, pp. 63-68.