Using LoRa Technology for IoT Monitoring Systems

Christos Bouras[1,2], Apostolos Gkamas[3], Vasileios Kokkinos[2], Nikolaos Papachristos[2]

[1]Computer Technology Institute & Press "Diophantus", Patras, Greece
[2]Computer Engineering and Informatics Dept., Univ. of Patras, Greece
[3]University Ecclesiastical Academy of Vella, Greece

bouras@cti.gr, gkamas@aeavellas.gr, kokkinos@cti.gr, papachristosn@upatras.gr

***Abstract*—This paper presents technology comparison scenarios for Internet of Things (IoT) concepts on rescue monitoring. The study starts by comparing WiFi & LoRa as wireless technologies able to be used by smart devices for data transmission. The IoT end-devices used in these concepts have high requirements in battery saving and for this reason the usage of low-power modules is advisable. This paper focus in rescue monitoring and the goal in the current study is the usage of the two wireless technologies used for data transmission from IoT devices: the already known WiFi and the upcoming LoRa technology. During rescue monitoring, important concepts are the identification and individuals' rescue of particularly vulnerable groups or individuals belonging to population groups with a high probability of being lost. A LoRa based gateway and WiFi Router is used to connect the end-devices used in our scenarios to the Internet. The collected data on server application as captured from installed sensors on the IoT modules can be displayed to authorized users through a web or mobile application. The results through simulation and real time experiments indicate that LoRa could be an ideal candidate for rescue monitoring. This study is a first step in creating a more general ecosystem for rescue concepts including all the hardware and software using the LoRa technology as transmission method.**

Keywords—IoT; wireless network; lorawan; narrowband; safety application; wearable sensor nodes;

I. INTRODUCTION

The Internet of Things (IoT) will prevail in many areas of our everyday life. The rapid increase of embedded IoT devices has introduced industries, companies and individual consumers to the development of added value IoT applications. People rescue is an important service sector for overall development and has far reaching a great impact on the quality of living. One of the most crucial problems is getting emergency response of illness and critical accidents in vulnerable groups of people that are possible to fall down sometimes or get lost. This concern may lead them to panic or lose their orientation or even to feel unable to respond to their everyday activities. A proposed wearable device is considered to be used in monitor scenarios. The 49% of people with autism spectrum disorders has been reported that they have disappeared or been at risk due to a tendency to flee at least once since the age of 4. Moreover, people who suffer from some form of dementia which have at least 60% possibility to get lost in outdoor areas. Finally, infants and children when exposed to large outdoor locations are likely to get lost. They are also extremely vulnerable to malevolent attacks and they are often unable to defend themselves. Real time monitoring of end-devices for example wearables can give us a clear overview for the current position of this vulnerable group of people. The IoT based solutions for such monitoring concepts through WiFi have been proposed in several works. In [1] for example, a comprehensive research of IoT monitoring has been presented. It provides several potential architectures of IoT for effective healthcare service provisioning. An energy efficient and trustworthy healthcare IoT system is proposed in [2], where the authors have emphasized on the security and energy consumption aspects of IoT based healthcare system. In [3], a body sensor network has been proposed for monitoring several health parameters. A similar approach has been proposed in [4] which is related to the health monitoring using a group of connected sensors to visualize people's health characteristics to central gateways for further consideration. In [5], a low power sensor design methodology has been proposed for mobile based healthcare applications.

Wearable IoT devices (end-devices), are generally deployed in wireless networks to monitor different parameters, such as user's position, current physical status, RSSI signals and alerts in case of emergency conditions. To allow long-term operation, these devices must be low power consumption and adopt energy harvesting as well as able to transmit their various captured metrics on long distances based on the available networks. These systems can also be installed on a bicycle, a car, animals or employees moving inside a factory or workspace acting like smart tags. In this research work, we present and study an IoT-based monitoring architecture able to be used for people rescue using LoRa and the traditional WiFi. Then we describe and analyse our design and implementation of the LoRa ecosystem platform suitable for health monitoring as well as simple test-cases related to our platform as a proof of concept for the data transmission [6].

The rest of this work is organized as follows: We briefly discuss the general system architecture of our ecosystem in Section II. Section III includes the use cases and experiments and Section IV concludes our hypotheses and discusses the future work.

978-1-7281-4445-0/19/$31.00 ©2019 IEEE

II. SYSTEM ARCHITECTURE

The requirements of a wireless IoT ecosystem are long battery life, with a minimum of 10 years of battery duration for simple daily connectivity with small packets size, low cost IoT module, low deployment cost, through limited new hardware installations and site visits and full coverage both indoor and outdoor. The proposed system in our study is the creation of a more general ecosystem for rescue concepts including the hardware and software. The implemented system will offer the possibility of basic communication with base stations that can be located many kilometres away from the end-device's location using two technologies. The first one is the traditional WiFi in which the end-device connects to a public WiFi Router and publishes its position and the current collected metrics (e.g. temperature & moisture) to the Things Network console (TTN), a set of open tools and a global, open network to build IoT applications. The second one allows connecting an IoT module using LoRa, acting like a real wearable device, to TTN using a nano-gateway. As we can extract from the flowchart below the end-devices' method of communicating with the base station and visualizing the data is a process consisting of checking network availability, capturing metrics, creating a transmission channel, and sending the data in the appropriate format so that can be exported from the cloud and visualized (Fig. 1).

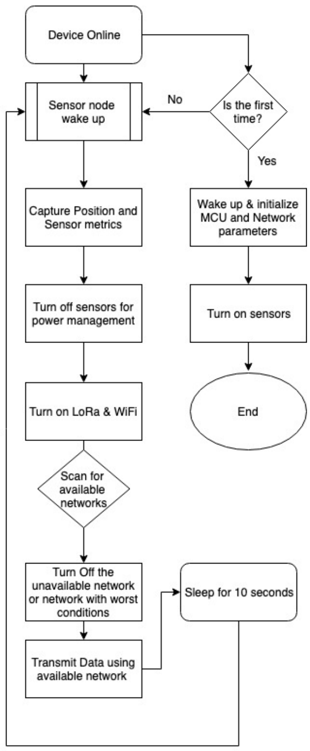

Fig. 1. Wearable node's software algorithm

For the implementation of our prototype we select the FiPy development board from Pycom[1] while there was a need for an effective energy harvesting module that can solve both energy and long-distance transmission issues. The FiPy is low cost (less than 20€ per node) and supports multiple technologies WiFi, LoRa, Bluetooth & NB-IoT in a unique module. Its architecture is based on the Espressif ESP32, with 512KB RAM and 4MB flash, RTC, plus GPIO. It is very popular due to its support for development via MicroPython[2]. MicroPython was the first Open Source scripting language ported to ESP32. For the calculation of the current position a Pytrack sensor shield which can be used with any of the Pycom multi-network modules is installed[3]. Some of the Pytrack features are the accurate GNSS Glonass GPS able to calculate the current position of the module, the 3 axis 12-bit accelerometer as well as the Ultra-low power operation (~1uA in deep sleep). For the temperature calculation a DS18B20[4] temperature (digital) sensor using 1-wire protocol has been installed. The above board gathers the data from various inputs (GPS, sensors values) and pushes them using our API on MicroPython to TTN using central gateways (WiFi or LoRa). In the first case, FiPy acts like an end-device (wearable) sending the collected data as JSON format to the UdiBots central webserver where the data are optimized through a friendly WebUI. For the LoRa case two FiPy Pycom modules used, the first one acts as a nano-gateway together with a LoRa Antenna suitable to send the data to TTN and the other one as a wearable end-device transmitting its position in time intervals to the above nano-gateway[5].

III. EXPERIMENTS AND USE CASES

a) End-Node connected to LoRaWan Nano-Gateway

The scenario is implemented using 2 FiPy development boards for nano-Gateway and end-node. Both devices are connected through USB cables for the current study and not by using battery support because the power consumption and battery life will be object of the future work. In the above evaluation, the end-device, after the successful connection to the LoRa nano-gateway, captures the current position and temperature and post them to the connected UDP stream in predefined time intervals. To verify the stability of this network for research purposes, indoor experiments were conducted. The gateway from the other side give access to the end-device connect to the internet, denigrate and merge the collected data from the UDP socket and posts its longitude, latitude, modulation, coding rate, SNR and RSSI value to TTN cloud server for further processing.

Fig. 2 shows the current position of the installed LoRa nano-gateway for our experiments. The algorithm's code of end-node as well as the LoRa nano-gateway is presented in the following pseudo code. In Europe, LoRaWAN operates in the 863-870 MHz frequency band. This is the Regional ISM band assigned for Europe only. The exact frequency value in our scenario is 868.1 MHz where code rate has constant value of

[1] https://docs.pycom.io/

[2] https://docs.micropython.org/en/latest/esp32/quickref.html
[3] https://docs.pycom.io/datasheets/boards/pytrack.html
[4] https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf
[5] https://github.com/pycom/aws-pycom

4/5 using spreading factor 7 and bandwidth 125W. The data rate should be as fast as possible so as to minimize the airtime (56.6ms). SF7BW125 that represents spreading factor 7 with 125W bandwidth used in our scenario is a good place for research study as it consumes the least power and airtime. If would like to have more range, we can slowly increase the data rate based on scenario needs. TTN console uses the non-standard SF7BW125 to SF12BW125 for frequency 868.1 in Europe. The payload size remains constant to 20 bytes through the whole communication as the packet sent by the end-device is constant also.

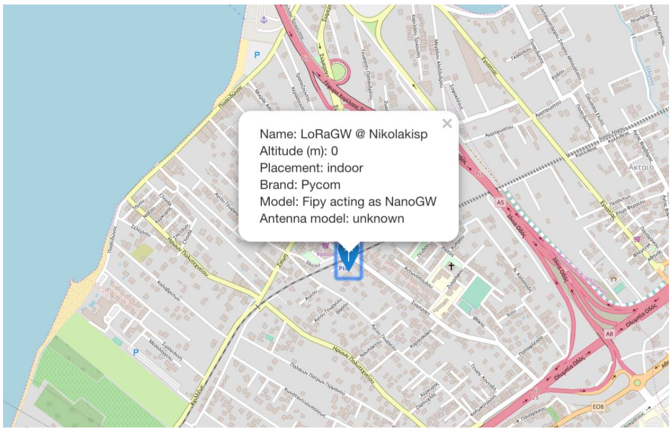

Fig. 2. Setup of the LoRa Nano-Gateway

Pseudo code of the LoRa End-Device

```
% Initialize LoRa in LoRaWAN mode in Europe
lora=LoRa(mode=LoRa.LORAWAN,region=LoRa.EU868)
    % Create OTAA authedication params for the Device End-
    node: DEV_EUI (Device Unique ID), APP_EUI
    (Application Unique ID), APP_KEY (Application Unique
    ID)
dev_eui = ('DEV_EUI')
app_eui = (' APP_EUI')
app_key = ('APP_KEY')
% Join LoRa network using OTAA (Over the Air
    Authentication)
lora.join(activation=LoRa.OTAA,auth=(dev_eui,app_eui,app
    _key), connection_timeout = 0)
% Wait until the end-node has joined the network
while not lora.has_joined():
  % Create a LoRa UDP socket and collect the metrics
  s = socket.socket(socket.AF_LORA, socket. SOCK_RAW)
  payload = {'lat' : get_curr_lat(), 'lot': get_curr_lot()}
  % Input DS18B20 library data line connected to pin 10
  ow = OneWire(Pin('P10'))
  temp = DS18X20(ow)
  payload = {' temp temp}
  s.send(payload)
  % Wait for the ACK after the successful data transmission
```

LoRaWan specifies a number of security keys: NwkSKey, AppSKey and AppKey. All these keys have a length of 128 bits. These keys are used to check the validity of the messages. In the backend of the TTN this validation is also used to map a non-unique device address (DevAddr) to a unique DevEUI and AppEUI. To support low-power mode in LoRa scenario, we used the power sleep function, and it is configured to periodically wake up for network availability discovery. For simplicity reasons we set up some basic LoRa parameters such as spreading factor, bandwidth and code rate to exploit the above network and take advantage of its capabilities on low power.

Pseudo code of the LoRA Nano-Gateway

```
% Setup LoRa GW parameters
init_network( ssid , password , server , ntp_server =
    'pool.ntp.org', ntp_period)
% Connect to the TTN cloud server and start nano-gateway
ntp_sync(ntp_server, update_period = ntp_period)
LoRa(mode, frequency, bandwidth, coding_rate)
% Create a UDP socket and wait for connection of end-nodes
s=socket.socket(socket.AF_LORA, socket.SOCK_RAW)
% LoRa radio events callback handler
if events & LoRa.RX_PACKET_EVENT:
  rx_data = self.lora_sock.recv(256)
  packet = make_packet(ex_data, rx_timestamp, rssi, snr)
  push_data(packet)
```

The received packet from the nano-gateway is decoded in json format. An example of such a data packet is given below:

JSON Decoded Data


```
{"frequency": 868.1,"modulation": "LORA",
  "data_rate": "SF7BW125","coding_rate": "4/5",
  "longitude": 23.7127,"latitude": 37.9667,
  "temperature": 35.2,"format": Celsius,
  "gateways": [ {
   "gtw_id": "eui-30aea4fffe78f320",
   "timestamp": 52467208,
     "time": "2018-12-15T16:35:01.409984Z",
     "channel": 0,
     "rssi": -111,
     "snr": 3
  } ] }
```


Each packet sent to TTN can then be visualized and used by the API for further processing [7], [8].

b) End-Node connected to WiFi Access Point

This scenario is implemented using 1 SiPy development board with WiFi, Bluetooth and SigFox connectivity. In the first case we start by calculating the current position and temperature as captured from the installed sensor and publish these values to the UdiBots Server. We used UdiBots, as a server because it provides a friendly User Interface using simple dashboards based on the captured data created by user. An example of the selected data in the UdiBots platform can be found in Fig. 3 and Fig. 4.

The figures show the position through time of the end-device as captured and sent to the Udibots cloud server which refreshes in 10 seconds time interval. These data can be used for further processing and optimization in Web or mobile applications [9]. It is known that WiFi is limited in range as the end-node must be in a distance up to 200m in order to be able to transmit data to a connected WiFi Access Point. In addition, WiFi consumes a lot of battery. This might be acceptable for our day to day use, but it is a problem for many IoT use cases. For instance, in many IoT applications as well as rescue concepts, a long-range communication is needed. The LoRa technology seems to be a more ideal candidate for this purpose as it uses very little battery, recognized as a low power consumption protocol. Also, it is an open specification, so anyone is free to implement the protocol on their own equipment as we already did for the gateway.

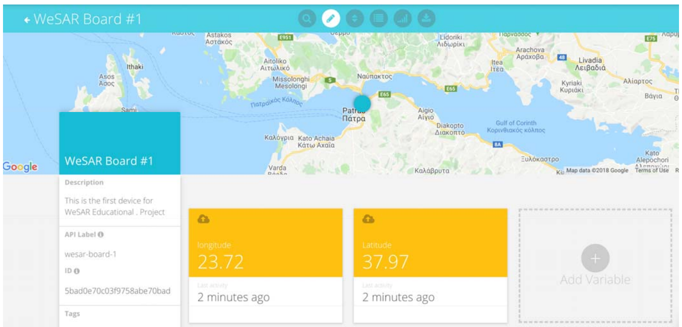

Fig. 3. Current position of the wearable device visualized in a real-time map based on the captured measurements.

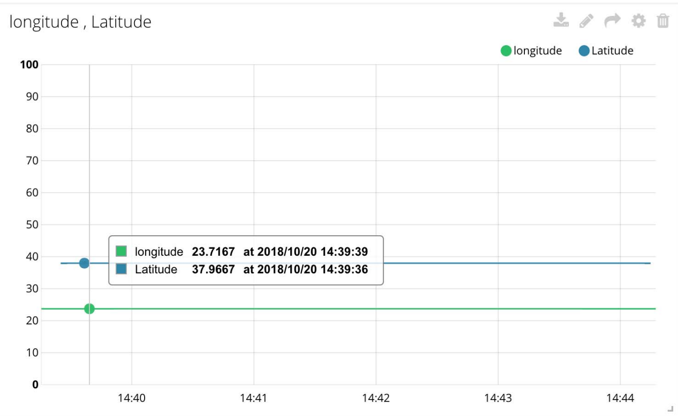

Fig. 4. Example of the current position of the IoT device through time

IV. Conclusions and Future Work

Both LoRaWan and WiFi technologies seem promising solutions for IoT monitor concepts. LoRaWAN is ideal for modules with sensors that only randomly send a value for example the position of an end-device every 10 minutes. It is also a good option for tracking-monitoring vulnerable group of people having a LoRa-enabled wearable as end-device in a predefined area. Currently LoRaWAN is ideal for long range using low power but also low bandwidth communication. On the other hand, WiFi only works in areas where the devices and the gateway are in short distance. A module connected to a WiFi Gateway also, uses a substantial amount of power, so it is not ideal for battery operated devices like wearables in our scenarios. Future work in our study is the power consumption minimization of the device during the data transmission and network switch based on the network availability.

V. Acknowledgments

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE - INNOVATE (project code: T1EDK-01520).

VI. References

[1] Y S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, K.-S. Kwak, “The Internet of Things for Health Care: A comprehensive Survey,” IEEE Access, vol. 3, no. 6, Jun. 2015.

[2] A. Sawand, S. Djahel, Z. Zhang, F. N.-Abdelsalam, “Toward Energy Efficient and Trustworthy eHealth Monitoring System,” China Commucications, vol. 12, no. 1, pp.46-65, Jan. 2015.

[3] K.-H. Yeh, “A Secure IoT-based Healthcare System with Body Sensor Networks,” IEEE Access, vol. 4, pp. 10288-10299, 2016.

[4] P. Gope, and T. Hwang, “BSN-Care: A Secure IoT Based Modern Healthcare System Using Body Sensor Network,” IEEE Sensors Journal, vol. 16, no. 5, pp. 1368 – 1376, May 2016.

[5] X. Chen, W. Rhee, and Z. Wang, “Low power sensor design for IoT and mobile healthcare applications,” China Communications, vol. 12, no. 5, pp. 42-54, May 2015

[6] Y. Tang and C. Li, "Wearable indoor position tracking using onboard K-band Doppler radar and digital gyroscope," 2015 IEEE MTT-S 2015 International Microwave Workshop Series on RF and Wireless Technologies for Biomedical and Healthcare Applications (IMWS-BIO), Taipei, 2015, pp. 76-77.

[7] G. Tanganelli, C. Vallati and E. Mingozzi, "CoAPthon: Easy development of CoAP-based IoT applications with Python," 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, pp. 63-68, 2015.

[8] A. Lavric and A. I. Petrariu, "LoRaWAN communication protocol: The new era of IoT," 2018 International Conference on Development and Application Systems (DAS), Suceava, 2018, pp. 74-77

[9] J. Joshi, P. Medikonda, R. S. Raja and D. Das, "Sub-gigahertz technologies based fast message delivery in VANETs," 2017 IEEE 15th Student Conference on Research and Development (SCOReD), Putrajaya, pp. 269-274, 2017.