A Novel LoRa-Based Wildlife Tracker

Lorenzo Albani
Delft University of Technology
Delft, The Netherlands
E.J.Kooij@student.tudelft.nl

Eric Kooij
Delft University of Technology
Delft, The Netherlands
L.Albani@student.tudelft.nl

***Abstract*—The research on wildlife using trackers is ongoing for more than ten years. Technology has since ten made its progress, now the LoRa technology seems to be a serious candidate in the IoT community. Using the available technologies will enable to create more reliable trackers which weigh even less. These lighter trackers enable to study the behaviour of smaller animals, even birds. The aim of the research conducted was to prove that such a system can be created using the present technologies, which resulted in a system containing a tracker and a user application.**

***Index Terms*—Internet of Things (IoT), Wireless Commmunication, Wide Area Network, LoRa, LoRaWAN, The Things Network (TTN), GPS, Localization, Logging.**

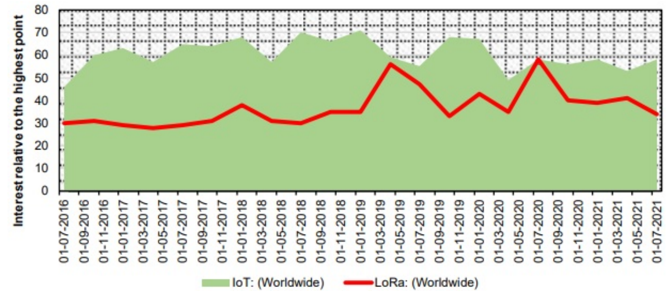


Fig. 1. Search statistics on IoT and LoRa [3]

I. Introduction

In the early 1960s the interest in biotelemetry started by the advent of VHF communication, which was replaced a decade later by satellite communication. Fine-scale localization was enabled by GPS technology starting in the 1990s, which was later extended by Geographical Information System (GIS) layers on which the biotelemetry still relies [1].

Nowadays the biotelemetry tracking data helped making decisions in ecological issues, therefor the scientific and commercial demand is still growing [1]. The biotelemetry tracking devices are a subsection of the Internet of Things (IoT) in the need to send data packets containing the location over a long range using a small amount of power, as its power is supplied by a battery, often combined by a solar panel.

Likewise to any smart object, a wildlife tracker would make use of a sensor (GPS) generating the data being monitored (Location) and a communicator to provide such information to the user. For years, the role of the communicator was embodied by a SIM Card on Chip relying on Mobile Networks.

The need for the low power communication over the long-range gained attention in the industry leading to the development of Low Power Wide Area Networks (LPWANs) as some active technologies are SIGFOX, LoRa, INGENU, TELENSA and QOSIO [2]. Amongst these competitors is the LoRa technology gaining the most interest from the community as can be seen in figure 1.

The aim of this paper is to describe the research performed during the creation of a tracking device using the LoRa technology. The tracker has the goal to be able to track wildlife, especially birds.

As LoRa communication will be implemented via The Things Network (TTN) ecosystem, this LoRa-based implementation of a tracker will significantly reduce implementation and experiment costs as the use of TTN is free of charge.

The outline of this paper is as follows, at first the LoRa & LoRaWAN technology is explained as it is the backbone of the tracking device and a relatively new technology as well, the explanation can be found in section II. In order to make sure the research does not duplicate others the results of related work can be found in section III. Furthermore the system design of the project as well as the experiments and their results can be found in sections IV and V respectively. The research will be concluded and its limitations and future work will be discussed in sections VI and VII respectively. At the very last a word of thanks to the people supporting this research has been added.

II. LoRa & LoRaWAN Technology

In this section a brief explanation of the LoRa physical layer and the LoRaWAN MAC layer on top of it is presented.

A. LoRa

LoRa is a physical layer operating in the Industrial, Scientific and Medical (ISM) band, which is an open to all, free, band. Usually the frequency band within a communication system is by far the most expensive part of the system, for which LoRa gained interest from the international community. The Sub-GHz band is the band where LoRa operates in, where the allotted frequencies differ per country. Furthermore the country specific frequency plans put restrictions on the duty cycle as well. [3]

The LoRa modulation technique, according to the patent that Semtech owns since the acquiring of Cycleo in 2012, uses the Compressed High-Intensity Radar Pulse Chirp Spread

Spectrum (CSS) accompanied by integrated Forward Error Correction (FEC) that helps reducing the interference due to different data rates. The frequency of the chirp signal is varied over time in order to change the phase between the symbols, if the frequency changes slow it will even decode at 19.5 dB which is even below the noise level of the floor. In order to allow for long range communication LoRa also takes care of the recovery in case of transmission errors. LoRa supports data rates between 300 bps up to 37.5 kbps depending on the transmission parameters: Transmission Power, Carrier Frequency, Spreading Factor, Bandwidth, and Coding Rate [4].

Theoretically the Transmission Power for the LoRa modulation will lie within the range of -4 dBm until 20 dBm, however due to hardware limitations the range of use will be starting from 2 dBm up to 20 dBm. Now the duty cycle is defined as the maximum percentage of the time duration in which an end device is able to occupy a channel, as is the key constraint by unlicensed bands. Considering the Carrier Frequency, this can be programmed to be in the range of 137 MHz until 1020 MHz taking steps of 61 Hz. However the LoRa Chip limitation allows it to limit the range to start from 860 MHz. The width of the frequencies within the transmission band, the Bandwidth, is strongly connected to the data rate. An increase of the Bandwidth increases the data rate as well, the range for the Bandwidth lies between 7.8 kHz and 500 kHz. Regular settings for the Bandwidth are either 125 kHz, 250 kHz or 500 kHz. The Spreading Factor is defined as the ratio between the symbol rate and the chip rate and thus defines the number of chirps per symbol as 2^{SF}. An increase in the Spreading Factor doubles the transmission time, so the transmission rate is halved, and increases the energy consumption as well as the Signal To Noise Ratio (SNR) eventually. The set of available Spreading Factors varies depending on the geographical location, for example in the north of America the available range is 7-12 while in Europe the series is defined as 6-12. An increase in the Spreading Factor increases the range of communication at the price of a higher sensitivity. Within LoRa orthogonal Spreading Factors are used enabling the transmission of multiple packets over the same channel concurrently. During transmission LoRa has to handle bursts of interference which it will control by selecting a suitable Coding Rate. The Coding Rate can be set to either 4/5, 4/6, 4/7 or 4/8, of which the last one is the most robust setting. A higher Coding Rate increases the robustness at the price of a higher on air time of the packet. Communicating radios using different Coding Rates can still communicate since the Coding Rate is specified within the header of each packet. Within LoRa communicating devices are able to select the values of the parameters automatically using the Adaptive Data Rate (ADR) feature. The Adaptive Data Rate helps the saving of energy while improving performance. [2]

B. LoRaWAN

LoRaWAN is the Medcium Access Control (MAC) layer build upon the LoRa physical layer using the LoRa modulation technology of which the specifications are defined by the LoRa Alliance, a graphical representation of the LoRa stack can be found in Figure 2 [5].

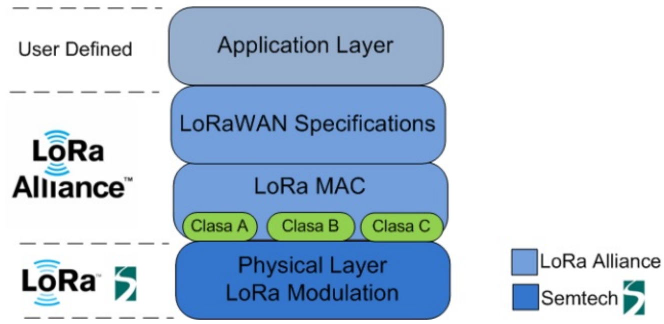


Fig. 2. A graphical representation of the LoRa stack [5]

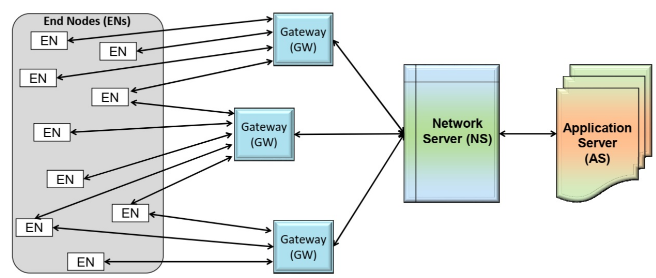


Fig. 3. A graphical representation of the LoRaWAN WSN [5]

The system architecture for a Wireless Sensor Network (WSN) has been standardized by the LoRa Alliance and so defines the LoRaWAN MAC communication protocol. LoRaWAN contains end nodes, also called motes, gateways, network servers and Application Servers, the architecture of a WSN can be found graphically in figure 3.

Within the WSN the end nodes are communicating directly to the gateway forming a star topology kind of network, furthermore the gateways form on their turn a star topology by their connection to the network Server for which reason the LoRa network is often called a 'star of stars' network. [5]

Regarding the end nodes the LoRa Alliance defines three classes namely A, B and C which the user will have to select based on the application needs. Class A end nodes are able to initiate a transmission at any time. After the transmission of the data from the end node to the gateway the protocol defines two receive (Rx) slots in which the end node needs to be able to receive data from the gateway, between the slots some delay is defined and either one of the two receive slots will be used. Since the end node decides to transmit it is optimal for battery powered devices which are in sleep from time to time and will transmit only when needed, a graphical representation of the class A communication scheme can be found in figure 4. Whenever there is the need for more frequent communication from the gateway to the end node class B can be used. Within class B the receive slots are scheduled periodically called beacons, the period between the beacons is referred to as the beacon period. After receiving the beacons the end nodes

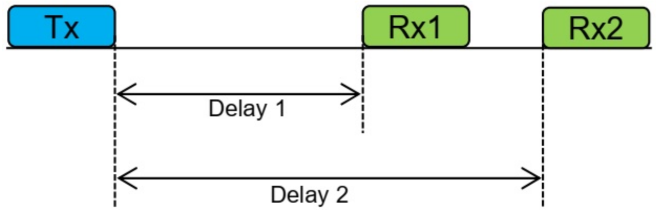


Fig. 4. LoRa class A communication [3]

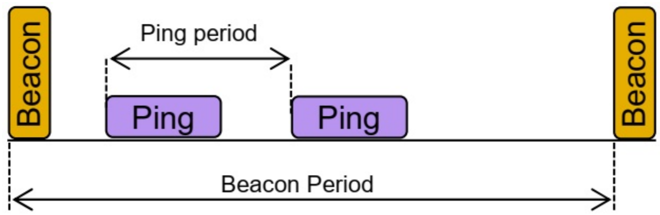


Fig. 5. LoRa class B communication [3]

will open receive, in class B also called ping, slots at the specified rate. Using the scheme of beacons and ping periods the gateway is able to synchronize the communication within the network at the cost of a comprised power consumption compared to class A, a graphical representation of the communication of class B can be found in figure 5. Finally class C are defined the devices which are constantly in the ”on” state and are except from the uplink communication in listening mode continuously. Class C devices provide low latency at the price of a high power consumption, for this reason the class C devices are rarely battery powered. A graphical representation of the class C communication can be found in figure 6. [3]

C. Pros & Cons

LoRaWAN is a communication protocol build upon LoRa which is able to send packets over a long range in the free ISM band. For this reason LoRa gained its interest, but also this technology is bound by its limitations. Furthermore it rarely happens that a technology does not contain its cons, in this section the pros & cons are stated. Using LoRa comes with the advantages as listed below:

- The operation happens within the free ISM frequency band
- Designed for low power consumption which enables an opportunity for battery powered devices

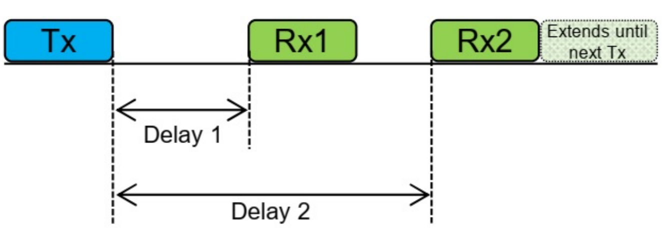


Fig. 6. LoRa class C communication [3]

- High security
- Simple architecture
- Good receiver sensitivity
- Documentation well written and widely available
- Large supportive community
- Firmware updates over the air are possible
- Readymade boards available, but the opportunity for customization is left open. [3]

The technology seems endless in the advantages, but one has face the limitations as well. The cons of the LoRa technology are listed below:

- Low bandwidth
- The technology might require trained personnel to operate and maintain the technology based applications. [3]

With the knowledge of the the advantages and limitations one can implement this technology within their projects well informed, or not.

D. LoRaWAN & regulations in Europe

Using the unlicensed band comes with constraints and regulations as was discussed in section II-A. Since the tracker will be mainly used in Europe these constraints and regulations will be discussed in this section.
Usually devices transmit based on a simple ALOHA based multiple access scheme and do not account for carrier sensing and/or Listen Before Talk (LBT) scheme. Devices using the European Sub-GHz band on the frequencies 433 MHz or 868 MHz are required to respect the strict regulations as specified by the European Telecommunications Standards Institute (ETSI). [11]
At first the duty cycle of 1 % has to be respected and the transmission power has to be limited to 14 dBm. Furthermore the Spreading Factor can be selected from the set $\{7\ , 8, 9, 10, 11, 12\ \}$, the bandwidth from the set $\{\ 125, 250, 500\ \}$ kHz and the Coding Rate from $\{\ 4/5, 4/6, 4/7, 4/8\ \}$. The values can be selected at design time, but are allowed to be changed during run-time. For the change of the parameters it is advised to use the Adaptive Data Rate (ADR) algorithm as specified by the LoRa Allience. The ADR requires both the network server and the end nodes to monitor the uplink quality and based on this update the parameters in order to improve reliablility and energy consumption. [11].

E. The Things Network (TTN)

The Things Network (TTN) creates networks, hardware devices and other solutions using LoRaWAN and is a global collaborative Internet of Things ecosystem. TTN is build upon the Things Stack Community Edition, an open, crowdsourced and decentralized LoRaWAN network. The TTN is a freely available network enabling the testing of devices, applications and integrations using LoRaWAN.
Upon registration, TTN enables for users’ own LoRa

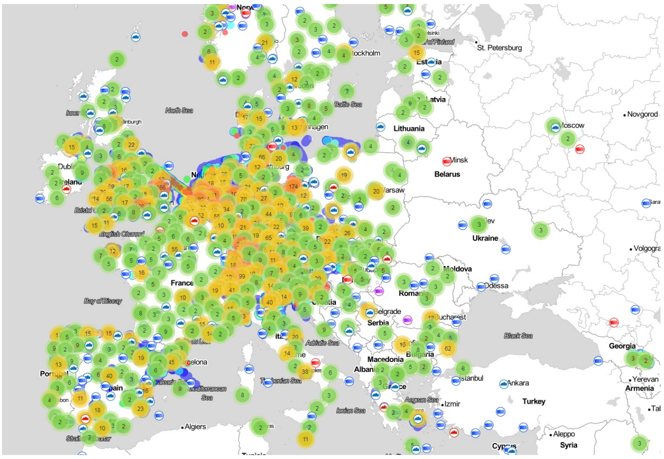


Fig. 7. European LoRa coverage of TTN [6]

End Nodes anywhere access to a global, volunteer, user-crowdsourced network of gateways to relay payloads to the internet: once an account has been created one user may add ones devices, gateways and integrations easily. Tools also exist to process the received Payloads. Furthermore it should be noted that commercial users will have to use the enterprise version. TTN is a global LoRa Network which means that locally the regulations are respected, the coverage within Europe can be found in figure 7 [12].

III. Related Work

Already in the year 2012 the authors of [9] realised by the imagination of no technological constraints that animal-attached tracking devices will unveil secrets, movement and behaviour of free-living animals in more detail then ever before. In the time of the research the trend of miniaturization of electronics was already ongoing continuously generating new developing opportunities. Furthermore the investigators realised that in order to obtain valid measurements the tracker should not affect the animal its behaviour, which effectively means that for a bird the device should not weigh more than 3 % of its body mass and should furthermore be small and optimized for aerodynamic shape. Once the bird carries the tracker there should not be the need for humans to touch the device anymore, even when the research is finished there should be a release function to drop the tracker from the bird its back. The tracker developed contained a GPS sensor, data storage, transmission and management as its main components. However LoRa was not available yet the team decided to use ZigBee transceiver and a base station instead, the range was extended if needed by an extended network of antennas.

With wireless technology improving in the year 2015 and the rapidly growing Internet of Things the group of [7] has developed to be able to monitor the water level. The inspiration for doing the project was to help cattlemen improve their troughs to become automatically refilled. The refill system reaches the underflow state when it fails to start filling in time, in the other hand when the filling system fails to stop in time an overflow occurs and only when the system operates normally the water level will stay within its bounds. At first the system needs to be able to know the water level in the trough, for this a monitoring system needed to be designed. Furthermore troughs are usually placed inside the barn without electricity, therefor the devices have to be battery powered. Furthermore sufficient transmission power is a must for the device since the hub and the barn are more than two kilometers away from each other. In the experiments the gateway was placed up on the roof of the cattlemans' house for the reason of the presence of a power connection. The troughs typically are within the range of one kilometer up to three kilometers around the gateway. From the start the sensor hub will broadcast a beacon to the air and the nodes will scan their surroundings for these beacons and select the one from which it received the highest RSSI. The node sends a request join packet on which it will receive its logical address as response from the hub. A node which has been unresponsive for more than 30 minutes gets marked as non-active and will be discarded after an hour of inactivity. For the node to become active again it simply repeats the join procedure once more. During normal operation the nodes send their measurements periodically to inform its state, which can be: underflow, normal, overflow or error. By having the system as described implemented the cattlemen are able to observe their livestock remotely which is a great improvement in order to do their job.

Even in the year of 2018 the field of maritime communications is still challenging according to the people who wrote [8]. The number of maritime services and operations that depend on communication to be trusted is large and rising. Since the maritime communication depends heavily on old techniques there is room for improvement, especially in the coastal communication field. In the coastal areas a fair part of the boats and ships is power limited as they are powered by batteries, for which reason the LoRa technology was selected as well as for its transmission range. The devices will contain the sensors it will need for its purpose, the authors take the Optimist sailboat as an example and follow these in training tracks within the harbour of vigo in Spain. In case of the Optimist device the following sensors were included: GPS, inertial, barometric, amperimetric and light intensity. Furthermore in order to meet the power limitations the Spreading Factor was set to 12, the maximum. The group succeeded in creating a system which shows a maximum transmission distance of four kilometers under the conditions of the deployed test-bench.

IV. System Design & Implementation

Using LoRa as the backbone of the system naturally adopts the LoRa architecture into the system, which will be further explained in IV-A. The architecture of the system consists of the End node(s) being the device(s), Gateways and a User application, which will be described in sections IV-B, IV-D and IV-E respectively.

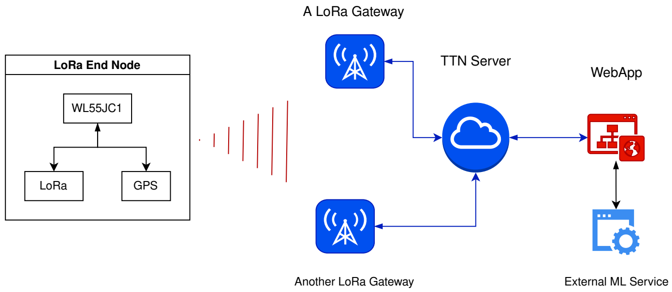


Fig. 8. The System Architecture. All arrows represent communication occurring via internet.

A. Architecture

The architecture of the system designed for and produced by this research is constituted by a LoRa End Node (Nucleo WL55JC1 Board + LoRa Radio + GPS Sensor), and a User Application. Each is described in detail in the following sections. Backbone of the system and an involuntarily actor to the architecture is also the Things Network gateway ecosystem and server stack, which is responsible for processing and relaying the payload received by the TTN gateways to the Web Application. The diagram in Figure IV-A illustrates the normal operation when the system is active: the GPS position and related data obtained by the GPS sensor is read by the NUCLEO WL55JC1 development Board and transmitted as a Payload using the LoRa Radio. All gateways receiving such payload will relay it to the internet to The Things Network Server Stack where it is available for the user to see and trigger automatic processing. A Webhook relays the data to a User Web Application where data is presented visually. An External Machine Learning Service is also communicating with the website in the cases described by Code 1.5 in Section IV-E2, product of the research further discussed in Section V-D.

B. End node

In order to get started using LoRa and being able to connect to The Things Network (TTN) the ST NUCLEO WL55JC1 [12] was provided. This board is a development board from the company ST to enable users to build prototypes with their STM32WL Series microcontrollers trying different combinations of performance, power consumption and features. The NUCLEO development board contains the STM32WL55JC microcontroller multiprotocol LPWAN dual-core 32-bit ARM Cortex, including 256 Kbyte Flash and 64 Kbyte SRAM memory, as its main component. In order to support the development the user is provided 3 LEDs, 3 buttons and a reset push-button. There are two oscillators installed on the board which are one 32.768 kHz LSE and 32 MHz HSE. furthermore the board is equipped with a USB type Mircro-B, a MPI debug connector, an expansion connector compatible with ARDUINO Uno V3 and a ST morpho extension pin headers which enables access to all STM32WL I/Os. The NUCLEO board is flexible in power-supply options, since it can be powered by ST-LINK, USB V_{bus} or even external power sources. Furthermore the board contains an on-board STLINK-V3 debugger and programmer with USB re-enumeration capability like mass storage, Virtual COM port and debug port. [13] Comprehensive and free software libraries, tutorials and code examples are available within the even free STM32CubeWL IDE. [12]

The Firmware coded for the device consisted in the reduction to the minimum of the use of the framework being implemented in the Application Examples of the IDE. Communication with the GPS was set up on top via Direct Memory Access (DMA) for the UART: purpose of DMA is to not interrupt any other process (Especially those regarding communication with LoRa as the Acknowledgement might be missed if Polling other devices for too long) whilst awaiting for data from the GPS. This is especially useful in case the GPS fails to transmit the position within a certain time. The GPS works by encoding the position and other location details via NMEA 0183 or simply NMEA Sentences. Only when GPS via DMA successfully fills the buffer with a sentence, an algorithm is triggered to process the sentence and update an object with last obtained position stored (if the NMEA sentence contains a position). Once processing is over, the DMA is made await again for the GPS to fill the UART RX buffer before triggering another sentence processing. Regardless of whether position is obtained, if connection with the GPS works, the device will persist in performing continuous connection attempts (Namely ”Join Attempts”) to The Things Network via OTAA (Over the Air Activation). If the LoRa Middleware flags a connection is established, the device proceeds to send a payload containing a position (which structure is discussed in Section IV-C) every 50 sentences received (this is done to implement some sort of duty cycle). It requires one gateway to receive the payload for it to be available on TTN (and thus on the internet) for further processing, later discussed in Section IV-D.

C. Packet structure

The Device firmware obtains the latitude, longitude, altitude, heading (in degrees) and speed (in km/h) extracted from the parsed NMEA Sentences produced by the GPS sensor. The five aforementioned values are packed in a 5, 32-bit (4 Byte) integers structure which can be conveniently read as a 20 Byte buffer starting from the struct pointer. The 20 Byte buffer is the payload transmitted to The Things Network. The header and tail of the packet is added by the Middleware as defined by the LoRaWAN specifications and subject of no change.

D. TTN Gateways Connection

The system relies on the significant, user-crowdsourced extent of coverage of The Things Network to enable the device to communicate its payload (e.g the position) to the internet. Upon registration of the device to the TTN User Console, any payload transmitted by the device and received from the gateways will be available on the platform for users.
TTN also offers processing tools (Payload Formatters) to make sense of the byte-encoded payload: it is possible for the user

to define a custom function to specify how the bytes in the payload should be decoded, and format the ”extracted” data into a more meaningful JSON string. On top of that, TTN also offers to trigger the formatter whenever new data is sent by a LoRa device and to relay the custom-formatted data to a web application, via user-defined Webhook.

A payload formatter has been implemented on the TTN User Console that would extract Coordinates, Speed, Altitude, Heading in degrees, and validate whether the data produced is meaningful. As the device has Big-Endian Byte Order implemented, the payload formatter also turns the value to little endian. Whenever a new payload arrives from the device, the formatter is executed and the output is passed to the Webhook that immediately delivers it to the Use Application via POST Request.

Also worth mentioning and extremely relevant for the research performed and extensively described in Section V-D is the fact the Webhook also relays the list of TTN gateways that received the payload, inclusive of each’s position and signal parameters.

E. User application

A User Application (or ”Web App” or ”User App”) has been developed and deployed on the web to enable users to interact with information received by the device and processed by TTN.
The web application offers a basic interface to track the device’s position in real-time, activate path recording and visualise previously (or currently being) recorded paths via a dedicated interface.

1) Technical details: The Web App has been developed using basic HTML+CSS elements for the front-end part, with a major effort concentrated on making the application a real-time interactive experience: users may navigate through all available sections and functionalities on the same web page, without requiring any page refresh to retrieve new data or performing any of the available operations.
This became possible by the use of Javascript (JS) intensively: JS would dynamically re-generate the page when either the user navigated the website or performed operations. In the latter case, JS code performs requests to the back-end part of the application coded in PHP (and executed by the web server PHP interpreter).
Back-end scripts are divided in those performing processes (Path Recording, Logging) and those that specifically care-take of processing newly arrived positions or fetching the latest from the database. The latter scripts are henceforth referred as ”connectors” and allow interaction with the database. It is through these connectors that the website updates each fixed interval of time the page with the latest data: connectors will receive a request from the JS code to check the database and return the latest position of the device for the website to dynamically update in the tracking page.
PHP connectors are also receive the POST requests containing the latest data from TTN (e.g. Device position), and proceed in interpreting the data (e.g. establishing if the altitude is valid) and store it into a MySQL/MariaDB Database.

If the user linked to the device has activated path-recording, the CSV file specified by the user is updated with the latest position after it is processed by a connector.
Both operations do occur regardless of whether the user is logged in or not.
If the user is online and in the tracking page, the user’s browser will execute the JS code, and the aforementioned operations of retrieving the latest position and updating the page will occur constantly, and shall continue doing so until the page is closed.

While the back-end part mainly exists to receive position updates, the most of time spent in back-end development was dedicated to enhance techniques to make sense of the data, in order to estimate a position when the device’s GPS is not able to provide one.

2) Website structure: The website constitutes of the following pages:

- **Live Device Tracking**
- **Saved Paths**
- **Wiki & How-Tos**
- **Settings**
- **Info & Credits**

In **Live Device Tracking** it is possible to follow real-time updates of the device position. When this mode is selected, the Web App will have the map pan to the latest position. On the right column, the site menu will be replaced by the device’s latest Latitude, Longitude, Altitude, heading information, and timestamps of the last received ping (e.g a connection) or valid position from TTN (and the device). A status message hints at the device connection status and the processes the data was subjected to: ”GPS LOCK” (Status Code: 2) would indicate a position was obtained by the device and ”PINGING” (Code: 0) would indicate the device communicates with TTN but no position was obtained either from the gateway list or the device itself. ”UNREACHABLE” (Any Code) would indicate that no updates from TTN were received recently, and the latest update was more then a minute ago.
On top of these cases, two kinds of ”NO GPS” status messages will both indicate that the GPS on the device did not obtain a fix position (just like PINGING) but TTN has given the position of at least one LoRa Gateway that received the message and either it was the case that only one Gateway position was obtained (Code: 1, Gateway position gets displayed) or signal data from 3+ Gateways was received and successfully used to triangulate or estimate a device position (Code: 1.5). Triangulation when no GPS data is available is possible thanks to significant research work allowing to implement a Machine Learning model on an external service, further discussed in Section V-D.

In addition to monitor the status, in this section is also possible to activate Path Recording on a specified (new) CSV File: this will trigger a change in the database that will ”inform” the application to store the newly received data on that CSV whenever new data arrives and will work even if the user closes the page. The CSVs linked to a username can be displayed by that user only on the Saved paths page, even if the displayed path is being recorded still in the while.

Additional touches for the tracking system is the emulation of a Google Maps functionality for navigation that will automatically follow the arrow in the map when the position is updated, unless the user moves the map: to re-follow the arrow's (and the device's) movements upon new updates, the user will have to click on the centering button.

In **Saved Paths** it is possible to display data related to a recorded path (even being recorded). The website only loads in the list of paths to choose from the CSV files linked to the user account.
When a CSV is selected, the path will be loaded from the file and displayed in the map. The file is also displayed in the right column in an interactive table: the user may click the table rows and the map will display the position related to that recorded entry in the CSV file.
At the end of the table, a download link exists to enable users to process the data recorded on CSV with any software or personalized script they would rather prefer.

In **Wiki & How-Tos, Settings and Info & Credits** sections can be found respectively instructions and details on the project, settings for the web page, and a short Credits note for visitors that may have encountered this website in the internet by chance.

V. Experiments & Results

In order to create a working tracker several experiments have been conducted. This section includes a description of the experiments as well as the results obtained.

A. Connecting to the network

Compared to Mobile Networks, the use of LoRa as the communicator for a GPS tracker has significant challenges: Range of LoRa communication can be significant if not obstructed. As most of the initial prototyping of the device was conducted indoor, this constituted a severe issue that prompted constant changes on the setup, slowing down the progress.
Significant time was spent on picking the Activation Type: as Activation By Personalization (ABP) mode guaranteed better changes the device would connect faster, Over-The-Air Activation proved to be more reliable after the network was Joined. Experiments were also conducted on the MAC Type and found that LoRaWAN 1.0.3 version resulted less needy (in terms of conditions) of further versions.
Experiments also had issues as the hardcoded implementation of the Duty Cycle in the Board blocked all the experiments after 2 minutes if no connection had occurred yet. This is further discussed in Section V-C

B. Obtaining the GPS data

The GPS was connected to the Nucleo Board via UART using DMA. No issues were registered in maintaining the connection, however buffering and parsing had to be manually implemented. Some challenges existed in obtaining GPS Data indoor that slowed down some experiments and casted doubts on the reliability of the setup.

C. Reliability

The concept of reliability for this device is twofold: first is the ability of the device to constantly transmit a payload whenever conditions that allow it exist and the second is the ability of the device to continue working up until the power is removed. The Nucleo WL55JC1 Board given contained a ”Framework” that was meant for the user to preserve and make use of to implement the desired behaviour on the board: this was cause of severe delays and forced to spend significant time understanding the Application Framework and how it interacted with the LoRa Middleware.
The produced simplified implementation clashed at times with such framework, which required constant work to ensure that the device would keep functioning in the various conditions (or states) it was expected to assume and especially when shifts occurred in such conditions.
In addition to that, significant time was spent testing how fast can the device re-connect to TTN, and if it is resilient (Both the System and the ability to Re-Connect) to sudden disconnections/reconnections from TTN. The result of this research was to implement the intended functionality by bypassing the application level functions, in addition to introduce a self-triggering self-reset whenever the device returned an error code given by the Duty Cycle implementation and whenever 10 minutes had passed since the last reset. Research showed that the memory the device retained of last LoRa connection was sufficient enough to quickly re-establish a connection after the reset.

Upon the implementation of such changes, the device was finally declared ”resilient” as per the definition given as it continued working uninterruptedly for more than 72 hours, during which it continuously broadcasted its position. The device was also placed on public transport and demonstrated ability to track moving vehicles.

The research for resiliency could also be expanded with the one conducted in exploiting LoRa on top of GPS to obtain a position: during the mentioned 72 hours, it was possible to collect data for experiments bound to overcome the problem of producing a position whenever the GPS is not able to (or to forgo the use of GPS), endeavor which is reported in the following Section V-D.

D. Research in Defining a LoRa-Based Localisation system

In some indoor, sky-covered location, the GPS has demonstrated to not be resilient enough in finding a position, with the exception of the case it retained memory of a previous, recent GPS Lock. This is because the data downloaded on the GPS Module while it was able to achieve

a Position is stored on a chip and may then be used to refine data and so derive a somehow exact location even indoors when line-of-sight (LOS) link is lost, it cannot be stressed enough that this occurs provided that such LOS GPS lock occurred first, and for significant time.
As the purpose of this device is to track locations of birds, such problem is not expected to be as frequent as during the testing phase, given the fact that a bird is reasonably expected to spend significant time in the air.
However, chances of GPS Chip Failures to obtain a fixed location are not reduced to 0, reason for which some sort of backup is believed required. In addition to that, the GPS Chip involves some significant power consumption: if it may exist some system able to obtain a location sufficiently precise to satisfy the needs of this research, and good enough to render the GPS Chip useless, it would be a significant breakthrough.

By registering the device on TTN, the Web Application was able to receive the payload generated by the device, in addition to the list of gateways that received the payload correctly, each paired with its position, Receiving time, Received Signal Strength Indicator (RSSI) and Signal Noise Ratio (SNR) of the received transmission.
Some research was conducted to model distance from the gateway upon the available parameters, so that multiple distances from multiple gateways could be used to triangulate a position. For all the attempts here described, some issue existed that tampered with the efforts of deriving a models, with some of those issues being beyond responsibility of the authors of this paper:

1) Time-based distance modeling: For each gateway that received the payload, a receiving time under the key ”received_at” was given by TTN in the JSON-Formatted data transmitted via the Webhook to the Web Application, together with the decoded Payload generated by the device. In addition, another ”received_at” key contained the timestamp of the first time the payload was received by the network: this will be used as a reference when timesequencing the experiment. This initial attempt only used time as a reference as it was believed that Signal-related Parameters may be heavily affected by surroundings. The tests were performed in a position in which availability existed of two TTN gateways in a straight line, plus one (without a registered position) next to the device which will be used as a reference to generate a ”starting” timestamp.

Given 0 used as in notation to indicate the starting point of the payload's journey in space (e.g where the Device is), 1 the first gateway in line and 2 the second, it is possible to model such journey. With $RXTime_1$ being the term indicating the time required to relay a payload on the TTN network (e.g the ”received_at” key sent by TTN) by the first gateway in a line from the moment it was sent, $Time_{0,1}$ the time required for the first gateway in line to receive such payload from the moment it was sent, dependent by distance $d_{0,1}$ from starting point 0 to 1 and last P being the processing time of the payload (From being received to be in the network), it is possible to assume to model the receiving time as can be found in equation 1.

$$RXTime_1 = Time_{0,1}(d_{0,1}) + P_1 \tag{1}$$

The second gateway in line would be be relayed on the network at $RXTime_2$, resulting in equation 2.

$$RXTime_2 = Time_{0,2}(d_{0,2}) + P_2 \tag{2}$$

since the time required for the gateway to receive the message (Notice: the gateway, not the network) is related to the distance travelled, such term is expanded, resulting in equation 3.

$$RXTime_2 = Time_{0,1}(d_{0,1}) + Time_{1,2}(d_{1,2}) + P_2 \tag{3}$$

P_1 will not matter to $RXTime_2$ as obviously Gateway 2 will not wait for Gateway 1 to relay the message on the network first before starting such process by its own. It is reasonable to assume some persistency in the internet connection reliability in both gateways and hence to believe both to take the same average time for processing, hence resulting in $P_2 = P_1$. Given the knowledge from TTN of both $RXTime_2$ and $RXTime_1$, which may affirm under the hypotheses in equation 4.

$$RXTime_2 - RXTime_1 = Time_{1,2}(d_{1,2}) \tag{4}$$

The relation between $Time_{1,2}(d_{1,2})$ and $d_{1,2}$ is given by the speed of light (in the medium).
However, when time came to calculate whether this relation was justified by the speed of light, the time associated with the difference distance of 1 Km resulted to indicate that the aforementioned gateways were rather 3000 to 10000 Km distant when speed of light in the vacuum was used.
The hypothesis of $P_2 = P_1$ would be kept if considerations made on the medium the payload travelled were as such as to justify a more-than-a-thousand factor in difference, however, as per the definition of the speed of light as in equation 5.

$$c = \frac{1}{\sqrt{\varepsilon_r \varepsilon_0 \mu_r \mu_0}} \tag{5}$$

Given Vacuum Permittivity and Permeability being fixed, such results would be justified when $\varepsilon_r \mu_r > 10^6$, an example of that would be the signal travelling in either Iron or Conjugated Polymers. As such materials are far more dense and complex then air, the model appears even more irrational in practice. It is possible that significant difference exists in Processing time ($P_2 \neq P_1$) given to overheads, other unknown factors or the chance the Real-Time Clocks in one or both gateways not being properly synchronized.

2) Signal and Time based distance modeling: It is widely known that both the RSSI and the SNR are influenced by distance: models that relate the RSSI to distance are often subject of literature [10], reason for which the available documentation on the subject was intensively studied. Data gathering hence started with purpose in modeling the relation of RSSI and

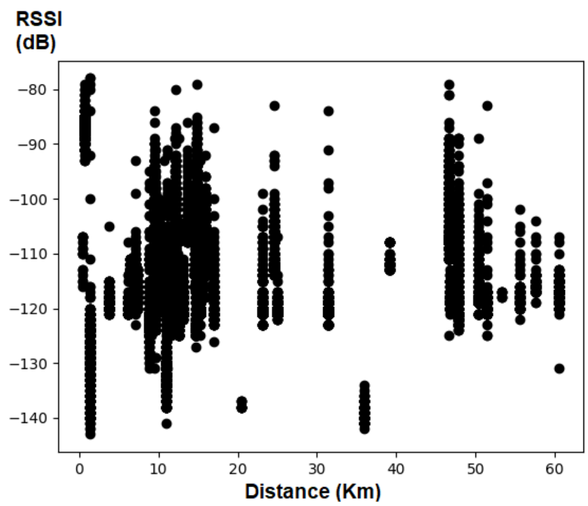


Fig. 9. RSSI over Distance, each point a Gateway.

SNR to known distances. Upon derivation of a model, RSSI and SNR would be used to predict the unknown distance of the device from the gateway(s). With the acknowledgement that signal is indeed affected by other non-distance related phenomena such as obstructions, the experiments were conducted on the 22^{nd} floor of the EWI Building of the Technical University of Delft to simulate coverage of an object (or animal) flying at high altitude, pending the approval of a request filed to access the building roof to repeat the experiment with better coverage which is likely to be interested by future work.

The device was made sending a payload every 20 seconds, and succeeded transmitting the payload 2555 times, having it received 72568 times over 72hrs. A resilience test also occurred while data was being collected. For each transmission, an average of 30 gateways succeeded in receiving and decoding the payload. The data sent by TTN (comprehensive of the payload and the gateways) was saved in a separate file for which a script was coded to process the data. Amongst the TTN gateways that received the payload, a significant subsection was far further distant than the known 10-kilometer theoretical threshold of LoRa, the most notable results being the payload received by a gateway on the top of a Bank in Utrecht, the port of Amsterdam and the furthest gateway being on top of the NPO Televisiontower in Hilversum, a significant 60.5 Km distance. For most of the distant results the coordinates did matche with a tower: given the flatness of the dutch landscape and the height of the building the transmissions started from, this result is justifiable by the towers' height not falling below the earth curvature: for a tower 60km distant, for instance, only 55m would be covered from a LOS transmission.

Upon processing, the RSSI, SNR, and Time required to get the message to a gateway would be put in relation with distance on a Graph and revealed evident issues with the experiment: as noticeable from Figure 9 and 10, hardly any visual relationship can be grasped from the RSSI and SNR over distance graphs, therefore forfeiting any attempt to model it mathematically.

On top of that, for a large portion of the gateways received, RSSI and SNR measurements "piled-up" on the scale: It is reasonable to believe that substantial portion of the gateways that received a transmission remained the same for each transmission. This generated additional challenges: in order for some useful values to exist such "piling-up" would have to occur mostly within a limited interval of SNRs or RSSIs for each gateway, thus allowing to define the rest outside the interval as "outliers" result of the rare appearance of unfavourable conditions. Given the presence of tens of thousands of points on the graph, this may be verified much easier by trying to use the data gathered to perform predictions, rather than waste time to manually scour for this interval by visually inspecting the graph. As a last attempt to make sense of the data, Machine-Learning based Regressors were employed to derive a model, using the RSSI and SNR as features. This revealed an average prediction error of 6.8 Km from the use of a Random Forest Regressor, being the lowest error in prediction registered if compared with the other two techniques tried being Support Vector Regressor (9.57 Km) and Multi-Layer Perceptron (7.12 Km). Attempts to further reduce the error continued with the introduction of time difference (between transmission and receiving) in the ways indicated in the section V-D1 as a third feature: this resulted in the Random Forest Regressor decreasing the error to 4.77 Km. As values began to fall below some acceptable threshold to obtain a sufficiently precise position (especially if the element being tracked is far enough for such precision to be significant), Time revealed to be a significant factor in "ordering" the data.

The last attempt to increase precision that was implemented in the model was filtering some invalid time measuring: as noticeable in figure 11, some gateways exhibited a receiving time that was several milliseconds in the past: this is clearly a problem given by Real-Time Clocks of the gateways not being properly synchronized. The graph discarded a significant amount of gateways reporting 1980 as the receiving year, which were also not fed to the Machine-Learning training. By removing the non-reasonable and negative time differences between sending and receiving, the average mean error had surprisingly increased: the hypothesis that might explain this is that the Random Forest Regressor is behaving as a Random Forest classifier, given the 30 discrete positions of the gateways that on average were reached by each transmission. It is likely that negative time differences were typical of a fraction of the gateways that could indeed be distinguished from the others from the mentioned negative numbers and so distance could be better predict.

Finally, given the apparent absence of an RSSI, SNR to Distance relation (on top of the considerations made on negative time), it is reasonable to assume, all factors considered, that the model produced might be right for the conditions (Placing, view, height) that it generated data from. In fact, the model produced had still sufficiently decent prediction results, and so was implemented as an external service running

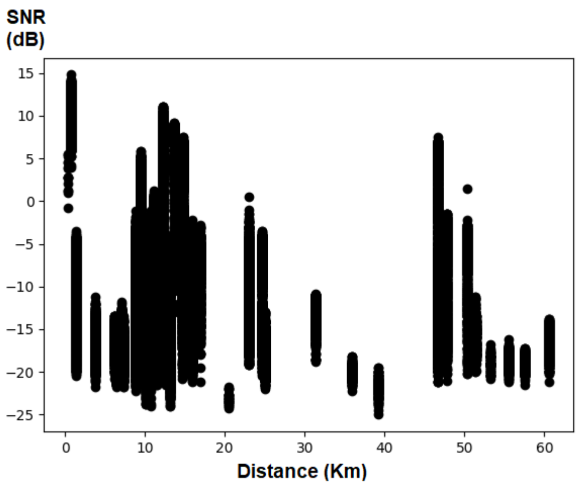


Fig. 10. SNR over Distance, each point a Gateway.

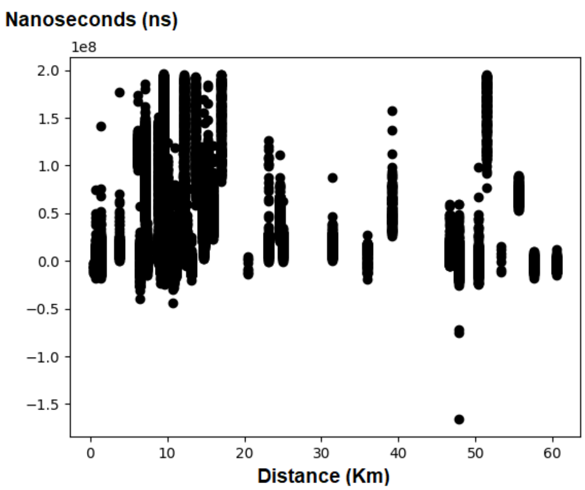
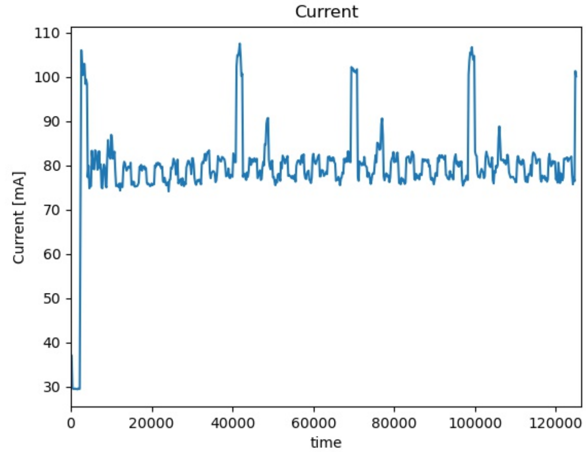


Fig. 11. Time in Nanoseconds Required over Distance, each point a Gateway. Notice the error registered.

on a RaspberryPi that is consulted by the Web Application whenever it is needed to perform such prediction, provided that three gateways are available with complete signal and timing data. The Model could not be run on the Web server containing the application due to service limitations of the host.

An additional mention should be given on the true final, yet ditched attempt to improve the model consisting in discarding all gateways that were beyond a certain distance. To compare the performance, a metric was established using the ratio of the most distant gateway to the average distance error: the higher the ratio, the better were the settings used. For data limited to gateways up to 13 Kilometers in range, this showed the impressive result of reducing the average error below the Kilometer, thus obtaining the maximum-distance-to-error ratio in prediction above 15.
However, as scarcity of TTN gateways is significant outside the Netherlands, not to mention outside europe, it may happen



f

Fig. 12. The current usage during operation of the device on the 20^{th} floor of the EWI building

to be useful that the model is trained to predict distance above the threshold that was set in this refining attempt, reason for which the decision to ditch this change in the model.

E. Energy consumption

As the ultimate goal of the tracker is to track wildlife, especially birds, it is important to design the system as energy efficient as possible. Ultimately the entire system operates solely on the energy provided by the harvester cooperating with a small battery or super capacitor.
In order to get insight in the energy consumption the application which has not been optimized for energy consumption was taken as the baseline measurement. The device was placed on the 20^{th} floor and on the 5^{th} of the EWI building of the TU Delft, during a period of operation the current usage was logged. The results of the measurements on the 20^{th} floor can be found in figure 12 and the measurements on the 5^{th} can be found in figure 13. The 20^{th} floor of the EWI building is higher than the surroundings and for this reason it does not encounter obstacles for the LoRa device, so it can be seen as an optimal situation. On the 5^{th} floor however the LoRa device encounters more obstacles and thus will have more difficulties in operation. The question now is how will the current usage differ between the two measurements.

Both figure 12 and 13 show a current usage of 80 mA when idle, the transmission of data takes a short spike of up to about 105 mA. The difference between the measurements can be found in the subsequent shorter spike that follows a transmission by 5-7 seconds, being the moment the device receives acknowledgements from the gateways. As the 20th floor tests resulted in more gateways receiving the payload, it is likely that higher receiving spikes are registered given the larger quantity of acknowledgements received.

VI. Conclusion

The aim of the study was to investigate the working and of a tracking device and create such a system using LoRa

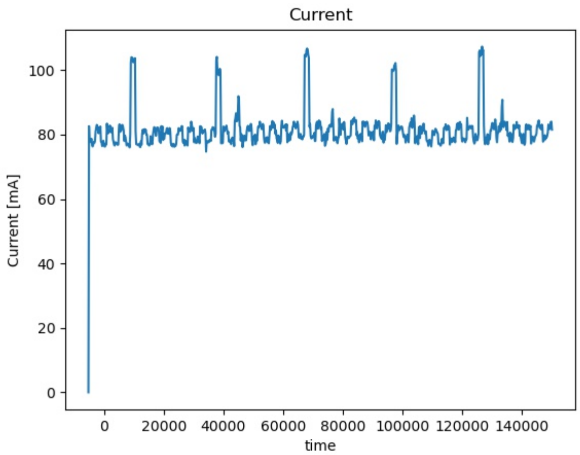


Fig. 13. The current usage during operation of the device on the 5^{th} floor of the EWI building

technology as it gains more and more interest from the IoT community. During the research a system design incorporating a tracker and a web application was created and several experiments were conducted. In conclusion of the research performed a working example was presented as a proof of concept.

VII. Limitations and Future Work

As a project carried out within the 'Advanced Practical I.o.T. and Seminar' course the duration of the research was very limited. The goal of the researchers was set to create a working and demonstrable system of a tracker and user application, which has been achieved.

The system created was designed to operate under the worst conditions and was not optimized for energy consumption at all, which would be considered the major step in the direction of an end product. Furthermore a custom designed PCB, methods of more accurate localization based on LoRa and a more extensive user application would be steps in the direction of an end product. At the point of testing on real animals it would become useful to be able to put the tracker to different modes and update the firmware over the air. Future experimental work may consist in updating the model produced with data either produced on the EWI Building roof or from the perspective of an actual bird. Later on the LoRa specifications and regulations in other parts of the world can be explored to eventually be able to export the product.

Besides the steps described the recent SuperGPS [15] should be taken into consideration for the long term.

Acknowledgment

This work has been conducted within the practical part of the 'Advanced Practical I.o.T. and Seminar' course which is lectured by Dr. RangaRao Venkatesha Prasad. The course is popularly referred to as 'Advanced IoT' and is part of the specialisation courses Networking within the Embedded Systems Master which is part of the education programme of the Electrical Engineering, Mathematics and Computer Science (EEMCS) Faculty of the TU Delft.

The authors like to thank Dr. RangaRao Venkatesha Prasad for providing such an interesting course and his extensive support to the research. Furthermore the people of Zero Energy Development (ZED), especially Niels Hokke and Sujay Narayana, for providing this interesting project to the Advanced IoT course and their time, enthusiasm and support.

Zero Energy Development (ZED) provides IoT solutions business-to-business. ZED combines more than 30 years of research experience into wireless and bateryless solutions in order to harness the potential of the next generation of IoT applications. ZED specializes in customized solutions based on the clients needs. By harvesting energy from the surroundings more and more IoT solutions operate batteryless. [14]

References

[1] (Thomas) Robertson, Bindi & Holland, John & Minot, Edward. (2012). Wildlife tracking technology options and cost considerations. Wildlife Research. 38. 653-663. 10.1071/WR10211.

[2] H. U. Rahman, M. Ahmad, H. Ahmad and M. A. Habib, "LoRaWAN: State of the Art, Challenges, Protocols and Research Issues," 2020 IEEE 23rd International Multitopic Conference (INMIC), 2020, pp. 1-6, doi: 10.1109/INMIC50486.2020.9318170.

[3] G. P. Reddy and Y. V. Pavan Kumar, "Demystifying LoRa Wireless Technology for IoT Applications: Concept to Experiment," 2021 4th International Symposium on Advanced Electrical and Communication Technologies (ISAECT), 2021, pp. 01-06, doi: 10.1109/ISAECT53699.2021.9668500.

[4] H. U. Rahman, M. Ahmad, H. Ahmad and M. A. Habib, "LoRaWAN: State of the Art, Challenges, Protocols and Research Issues," 2020 IEEE 23rd International Multitopic Conference (INMIC), 2020, pp. 1-6, doi: 10.1109/INMIC50486.2020.9318170.

[5] A. Lavric and A. I. Petrariu, "LoRaWAN communication protocol: The new era of IoT," 2018 International Conference on Development and Application Systems (DAS), 2018, pp. 74-77, doi: 10.1109/DAAS.2018.8396074.

[6] The TTN Mapper - Heatmap https://ttnmapper.org/heatmap/

[7] Lukas, W. A. Tanumihardja and E. Gunawan, "On the application of IoT: Monitoring of troughs water level using WSN," 2015 IEEE Conference on Wireless Sensors (ICWiSe), 2015, pp. 58-62, doi: 10.1109/ICWISE.2015.7380354.

[8] Sanchez-Iborra, Ramon & Liaño, Ignacio & Simoes, Christian & Couñago, Elena & Skarmeta, Antonio. (2018). Tracking and Monitoring System Based on LoRa Technology for Lightweight Boats. Electronics. 8. 15. 10.3390/electronics8010015.

[9] Bouten, W. & Baaij, Edwin & Shamoun-Baranes, Judy & Camphuysen, Cornelis. (2012). A flexible GPS Tracking system for studying bird behavior at multiple scales. Journal of Ornithology. 154. 10.1007/s10336-012-0908-1.

[10] J. Wang, X. An, R. V. Prasad and I. G. M. M. Niemegeers, "RF-OPT: A Range-Free Online Person Tracking System," 2006 Pervasive Health Conference and Workshops, 2006, pp. 1-5, doi: 10.1109/PCTHEALTH.2006.361677.

[11] Pop, Alexandru-Ioan & Raza, Usman & Kulkarni, Parag & Sooriyabandara, Mahesh. (2017). Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks?. 10.1109/GLOCOM.2017.8254509.

[12] ST NUCLEO-WL55JC Data brief - https://www.st.com/resource/en/data_brief/nucleo-wl55jc.pdf

[13] The Things Industries - The Things Network - https://www.st.com/resource/en/data_brief/nucleo-wl55jc.pdf

[14] Zero Energy Development (ZED) - https://www.zed-iot.com/#/

[15] SuperGPS - https://www.tudelft.nl/citg/over-faculteit/afdelingen/geoscience-remote-sensing/research/projects/supergps