Outdoor Localization and Distance Estimation Based on Dynamic RSSI Measurements in LoRa Networks: Application to Cattle Rustling Prevention

Ousmane DIENG
dept. Computer science
University Gaston Berger
Saint-Louis, Senegal
dieng.ousmane1@ugb.edu.sn

Congduc PHAM
LIUPPA, University of Pau
Pau, France
congduc.pham@univ.pau.fr

Ousmane THIARE
dept. Computer science
University Gaston Berger
Saint-Louis, Senegal
ousmane.thiare@ugb.edu.sn

***Abstract*—In this paper, we propose a RSSI-based distance estimation scheme for localization of cattle collars communicating with long-range LoRa radios. Cattle localization is designed to prevent theft in livestock. The proposed solution decreases the cost of cattle localization by minimizing the number of collars with GPS and allows accurate localization of collars without GPS. We propose a RSSI-based distance estimation using real-time adjustment of RSSI-distance mapping taking advantage of communication between collar nodes and gateway. Log-distance path-loss model is also used as rescue when the map does not provide accurate correspondence. Experimentation results show the validity of the approach with highly accurate localization of non-GPS collars.**

***Index Terms*—Localization, LoRa, RSSI Distance Estimation, GPS, Cattle Rustling, Kalman filter, Path-loss**

I. Introduction

Internet of Things (IoT) is recognized as one of the most important areas of future technology. In Africa, it is becoming the most important solution for many sectors in rural and farming activities such as agriculture, livestock, fish farming and so on. In livestock, farmers are confronted with the devastating cattle theft phenomenon. In a previous work [5], we showed that cattle theft can be prevented using IoT solution and particularly using the Semtech's LoRa radio technology belonging to so-called LPWAN (Low Power Wide Area Network) network category more suitable for IoT deployment. In that work, we proposed a GPS-collar-based localization and alerting system. Although this GPS solution is energy efficient, it remains a costly localization way for livestock tracking with hundreds of cows. Hence, in this paper, we study energy and cost efficient LoRa-based localization solution for livestock tracking in wide rural areas.

There are some challenges of using LoRa for localization because in some way its novel features constitute weakness points to achieve accurate localization. In fact, even though existing localization techniques give some accurate results in traditional wireless networks (ZigBee, WiFi, Bluetooth,...), they are practically unviable generally for LPWAN and specially for LoRa. For instance, most of ranging-based localization techniques lack of accuracy because of long distances and low bandwidth. Angle of Arrival (AoA) solutions, for example, are very weak in accuracy in long range because errors may increase with distance from the anchor points. In the same way, Time of Arrival (ToA) technique in trilateration approach, where distances between a device and each anchor node are estimated through time of arrival, requires the use of precise clock to synchronize with the network. This implies additional communication overheads and higher cost, therefore not ideal for low-power and low-cost LoRa device. Fingerprinting technique is also difficult to perform in long range and outdoor conditions because the offline phase requires more effort and time to cover wide areas. These approaches are unsuitable in livestock tracking when animals are grazing in areas much larger than the considered area during the offline phase. However, Choi et *al.* proposed in [2] an outdoor fingerprint positioning algorithm in LoRa network using interpolation technique to complete zones of the service area that were not covered in the offline phase. The average accuracy was about 28.8m but the considered area was very limited in size: 340mx340m.

A widely use and promising technique for LoRa network is multilateration using Time Difference of Arrival (TDoA) as introduced in [4], [6], [10]. This method consists of finding differences in distance from each gateway, instead of anchor node, to the device by calculating time difference of arrival of a signal from a device to gateways. There is no need for a device to be synchronized with the network. However, it requires tight time synchronization of the gateways to achieve the desired accuracy – since radio signal travels about 300 meters in free space over a time duration of 1μs, a time-stamping of 0.3ns is needed to get a 10 meter level of resolution. LoRa Alliance™ relates in [3] that localization has been performed with TDoA method. To achieved meter-level ranging, each gateway uses GPS to provide nanosecond time-stamping accuracy. In addition to the very tight time synchronization constraints, the LoRa Alliance solution is based on licensed gateway and the multi-lateration solver is not open-source. Nevertheless, we can find in the literature some studies using the same technique combining with some filtering or machine learning methods that provide acceptable accuracy. In [10] Podevijn et *al.* achieved 200m of accuracy. Fargas and Petersen improved the accuracy to about 100m with an iterative algorithm and an outliers detection method [6]. Lam et *al.* [8] adopted RSSI-based method to study LoRa localization in very noisy environment. They proposed algorithms to handle background, blocking and multi-path noise but no clear accuracy assessment is given. Kalman filter and machine learning methods are also studied for LoRa localization in respectively [1] and [7].

Most of the aforementioned solutions for LoRa localization need a large number of gateways as anchor nodes therefore a quite large infrastructure is needed. In rural areas, private LoRa deployments are the most common scenario with generally a very small number of gateways due to cost and deployment constraints.

In this paper, we propose an RSSI-based distance estimation in LoRa networks for cattle localization when both collar with and without GPS are deployed. We propose an original solution with only 1 gateway and, in order to improve the distance estimation scheme, an adaptive RSSI-distance mapping algorithms is run between end-devices (the collars) and the gateway. The rest of the paper is organized as follows. Section 2 presents thoroughly our proposed scheme. Section 3 details experimental results. Section 4 concludes the paper and gives some perspectives.

II. Proposed approach

A. Problem formalism

A first solution to prevent theft in livestock has been established using LoRa-based collar attached to animal's neck that allows accurate localization with an embedded GPS [5]. This solution had some limits as herds can consist in about a hundred animals so full-GPS solution can quickly become very expensive. For these reasons, we propose a new solution in order to minimize the number of collars with GPS and improve localization of collars without GPS by using GPS-collars as mobile anchors. The main issues with anchor approaches are generally (i) how to optimally use anchor nodes to localize the other nodes and (ii) how to reach the minimum necessary number of anchors to provide good accuracy? The use of the LoRa technology and natural behavior of a herd of cows help for the second issue: generally in a grazing herd, two animals are never separated by more than 500 meters and therefore with a transmission range of more than 1 km with LoRa in an open environment, each node can communicate with all the other nodes of the network. Therefore, reaching at least three GPS collars is usually not difficult to provide good accuracy to localize the other animals in the herd. Addressing the first issue is more challenging especially as we chose localization methods based on RSSI distance estimation technique because it is more accessible and does not need additional hardware, thus reducing greatly the cost of the whole system. Hence, the question can be considered as follows: how GPS collars can help to build an accurate RSSI-based distance estimation scheme? The rest of the paper focuses on this issue and will consider cows with GPS collar as GPS-node and cows with no-GPS collar as Beacon-node.

B. RSSI-based distance estimation Phase

The method we propose for distance estimation is a dynamic and continuous RSSI-distance mapping mechanism. The distance estimation is constantly improved with a weighted RSSI-distance correspondence procedure to determine the level of accuracy of an entry in the RSSI-distance map.

1) Network setup and RSSI considerations: The mechanism we are going to describe here is based of a particular network setup composed of a set of GPS-nodes and a set of Beacon-nodes communicating with one gateway. GPS-nodes broadcast their messages so both the gateway and Beacon-nodes can receive from GPS-nodes (see Fig. 1). To explain better the RSSI considerations in this particular network setup, we use the following notations: *GW* as the Gateway, B_i the Beacon-node i, G_i the GPS-node i, $d_{i,j}^{RSSI(k)}$ the distance separating node i from node j at step k estimated from RSSI values, $d_{i,j}^{GPS(k)}$ distance separating node i from node j at step k obtained from GPS coordinates, and $R_{i->j}^{(k)}$ the RSSI of the packet sent from node i to node j at step k.

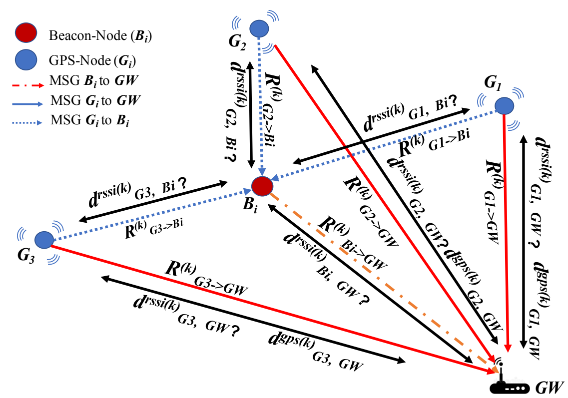

Fig. 1. Network communications with three GPS-node G_1, G_2 and G_3, one Beacon-node B_i, and the Gateway GW

Fig. 1 shows 3 GPS-nodes communicating with the Gateway (links highlighted in red color) and providing 3 RSSI-distance mapping based on GPS. Beacon-node B_i also receives these messages and will get 3 RSSI values: $R_{G1_i->B_i}^{(k)}$, $R_{G2_i->B_i}^{(k)}$ and $R_{G3_i->B_i}^{(k)}$. When B_i sends a beacon to GW, it will piggyback these RSSI values which will get at the same time $R_{B_i->GW}^{(k)}$. The RSSI-distance mapping based on GPS will help determining the distance from the Beacon-node to GW but also its distance to each of the GPS-node. Once these distances are estimated, it becomes straightforward to determine the position of the Beacon-node. Our contribution in this work is to propose a mechanism to continuously update and improve the distance estimation procedure. In addition, as animals are continuously moving in the grazing area, the number of RSSI-distance pairs based on GPS will increase, thus improving further the RSSI-distance estimation procedure over time.

2) Communication model: Nodes in the network communicate as depicted in Fig. 1. Messages exchanged between nodes will allow RSSI computation at the receiver side to enable RSSI-distance mapping. Two type of messages can be distinguished, those from a Beacon-node and those from a GPS-node. Beacon nodes unicast their messages to GW while GPS-nodes broadcast their messages so that both Beacon-nodes and GW can capture messages from GPS-nodes.

Communications follow a cycle of sleep and wake-up phases. We assume that all end-devices are synchronized for the sleep/wake-up cycle. As a tight synchronization is not necessary, it is easy to have a guard time for opening a received window based on a defined periodic wake-up value. We explain below the 3 steps of a communication cycle between a GPS-node G_i, a Beacon-node B_i and GW. These steps are repeated for at least 3 GPS-nodes.

1) **Step 1:** At each wake up from sleep mode all the B_i are listening for message coming from at least 3 GPS-nodes as illustrated in Fig. 2. In Fig. 2(a) G_1 broadcasts a message which is received by both B_4 and *GW*. Note that each GPS-node has a different transmission time to avoid packet collisions between GPS-nodes.

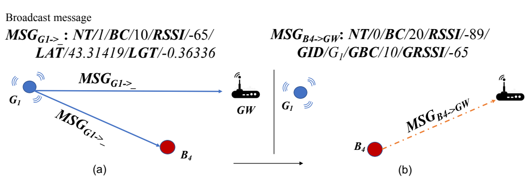

Fig. 2. Message sent by GPS-node G_1

2) **Step 2:** The Beacon-node that received the GPS-node's message will build and send a message to GW. This message contains the node type (***NT***), the Beacon Counter (***BC***), the RSSI of the message from the GPS-node (***GRSSI***), the id of the GPS-node (***GID***) and the Beacon Counter (***GBC***) of the GPS-node. This is illustrated in Fig. 2(b). Note that Beacon-node uses randomization when transmitting beacons to avoid packet collisions. Moreover, Listen-Before-Talk and CSMA-like backoff procedure is used to recover from packet collisions [9].
3) **Step 3:** The Gateway receiving these messages (from both GPS-nodes and Beacon-nodes) stores the different RSSI values and keeps a RSSI-distance mapping table with the estimated distance obtained from the GPS coordinates of the GPS-nodes and the Gateway.

3) Mapping and adjustment algorithm: According to the communication model, at each wake-up phase, GW first receives a packet from a GPS-node and can determine an RSSI-distance correspondence which will be used to update the RSSI-distance map. We consider M^{k-1} as the state of the map at wake-up phase $k-1$. Its format is M^{k-1} = [m_1^{k-1},...,$m_{N_{k-1}}^{k-1}$] where each entry m_i^{k-1} = ($<d_i^{k-1}, R_i^{k-1}>, w_i^{k-1}$). d_i is the distance, R_i is the RSSI associated with d_i and w_i is the weight of the RSSI-distance pair. N_{k-1} is the length of the map at step $k-1$.

At wake-up phase k, M^k is constructed by updating M^{k-1} with a new pair $p_k =< d_{G_i,GW}^{GPS(k)}, R_{G_i,GW}^{(k)} >$ built by GW with packets from GPS-nodes.

Algorithm 1 Weight-based RSSI-Distance Mapping

Input: Pair ($d_{G_i,GW}^{GPS(k)}$, $R_{G_i,GW}^{(k)}$) and M^{k-1}
Output: M^k
for all (c,w) in M^{k-1} **do**
  {c is a (distance,RSSI) pair and w its weight}
  if $c[1] = d_{G_i,GW}^{GPS(k)}$ **then**
    if $c[2] = R_{G_i,GW}^{(k)}$ **then**
      updateWeight(M^{k-1}, $d_{G_i,GW}^{GPS(k)}$, $R_{G_i,GW}^{(k)}$)
      {*updateWeight* increments the weight of the pair by 1}
    else
      {*addPair* adds a new pair to the map and initializes its weight to 1}
      addPair(M^{k-1}, $d_{G_i,GW}^{GPS(k)}$, $R_{G_i,GW}^{(k)}$, 1)
      {At this stage we can have in the map several pairs with the same distance but with different RSSI values}
      {*search* searches all pairs that contains this distance and return a list of all RSSI mapped with this distance}
      L_{RSSI}^d, L_{weight}^d = search(M^{k-1}, $d_{G_i,GW}^{GPS(k)}$)
      {λ is the minimum number of RSSI values judged enough to be filtered. The filtering process is done using Kalman method}
      if *length*(L_{RSSI}^d) = λ **then**
        F_{rssi} = kalmanFilter(L_{RSSI}^d)
        addPair(M^{k-1}, $d_{G_i,GW}^{GPS(k)}$, mean(F_{rssi}), mean(L_{weight}^d))
      end if
    end if
  else
    addPair(M^{k-1}, $d_{G_i,GW}^{GPS(k)}$, $R_{G_i,GW}^{(k)}$, 1)
  end if
end for

The algorithm updates M^{k-1} by performing the following actions:

1) Update the entry in M^{k-1} that contains exactly p_k by incrementing its weight by 1.
2) If M^{k-1} does not contain exactly p_k, add p_k to M^{k-1} and initialized its weight to 1.

As a result of action 2, for a given distance $d_{G_i,GW}^{GPS(k)}$ there may be several pairs in M^k due to the fluctuation nature of RSSI. Therefore M^k will be filtered to reduce the number of pairs. With $d = d_{G_i,GW}^{GPS(k)}$, L_{RSSI}^d = r_1,r_2,...,r_n is the list of the various RSSI associated to d and this list will be filtered using a Kalman filter. All entries where the RSSI is in L_{RSSI}^d will be purged in the map and replaced by a single entry where distance d will be associated to the mean RSSI computed on all the filtered RSSI values F_{rssi}. Noting L_{weight}^d as the list of the weights of all the purged entries, the weight of $< d, mean(F_{rssi}) >$ will be initialized with the mean weight computed on L_{weight}^d.

We chose the Kalman filter because its filtering process considers noise in data as a function of time. Therefore, a Kalman filter can efficiently handle the fluctuation (noise) in collected RSSI values. In addition, as L_{RSSI}^d contains only single value elements, the one-dimension version of Kalman filter depicted in Fig. 3 can be used.

The error in the estimation (e_E^0) and the error in measurement (e_M^0) are initialized with the standard deviation obtained from a pre-deployment phase (in this paper we use

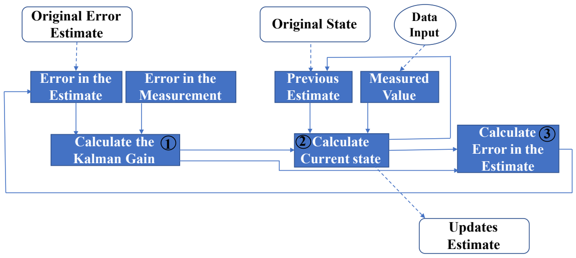

Fig. 3. One-dimension Kalman Filter flow chart

$\sigma = 1$ deduce from first experiment values, see subsection III-A) and the starting value is the first value r_1 of L^d_{RSSI}: $e^0_E = \sigma, e^0_M = \sigma, E^0 = r_1$. To filter L^d_{RSSI}, the three operations depicted in Fig. 3 are successively applied on all r_i. Then at each step i ($1 < i < n$, n being the size of L^d_{RSSI}) Kalman Gain G^i, Current State (current estimation) E^i and Error in the Estimate e^i_E are calculated using the following equations.

$$G^i = \frac{e^i_E}{e^i_E + e^i_M} \tag{1}$$

$$E^i = E^{i-1} + G^i(r_i - E^{i-1}) \tag{2}$$

$$e^i_E = (1 - G^i) * e^{i-1}_E \tag{3}$$

4) Distance estimation: Distance estimation at step k is performed when the gateway receives a packet from a Beacon-node. As indicated previously, a Beacon-node B_i can receive messages broadcasted from a GPS-node G_i and can therefore know the RSSI of the last broadcasted message, e.g. $R^{(k)}_{G_i->B_i}$. When B_i sends a beacon message to GW, it will indicate this RSSI value in the message payload. GW will then run algorithm 2 to determine the distance between B_i and (1) GPS-node G_i ($d^{(k)}_{B_i,G_i}$) and (2) itself ($d^{(k)}_{B_i,GW}$), using $R^{(k)}_{G_i->B_i}$ (in the message payload) and $R^{(k)}_{B_i->GW}$ (obtained from the packet reception).

This algorithm takes 3 parameters in input: M^k, $R^{(k)}$, and σ. M^k is the map at step k. $R^{(k)}$ can be either $R^{(k)}_{G_i->B_i}$ or $R^{(k)}_{B_i->GW}$. σ indicates which entry in the map has its RSSI component closest to $R^{(k)}$. As explained previously, the standard deviation of RSSI values determined in a pre-deployment stage can be used here as initial value. To estimate a distance from $R^{(k)}$, algorithm 2 finds in M^k the σ-closest entries corresponding to $R^{(k)}$. σ-closest is defined by Eq. 4.

$$\begin{aligned}\sigma - closest(R^{(k)}) = \{p_i =< (d_i, r_i)|w_i > / \\ p_i \in M^k, \sigma_i = |R^{(k)} - r_i| < \sigma, \\ \sigma_i = min\{\sigma_j\}, 1 < j < length(M^k)\}\end{aligned} \tag{4}$$

If the set of σ-closest entries has only one entry, the entry's distance is retained as the estimated distance and the accuracy of this estimation is the ratio of the entry's weight, w, on the number of occurrences of entry's distance in the whole map, N:

Algorithm 2 RSSI-based distance estimation and adjustment

Input: $R^{(k)}$ = $R^{(k)}_{B_i->GW}$ or $R^{(k)}_{G_i->B_i}$, M^k and σ
Output: $d^{(k)}_{B_i}$ = $d^{(k)}_{B_i,GW}$ or $d^{(k)}_{B_i,G_i}$ and updated M^k
  σ-closest= $\{p_i = <(d_i, r_i)|w_i > / \ p_i \in M^k$, σ_i=$|R^{(k)}$-r_i $| < \sigma$, σ_i=min$\{\sigma_j\}$, 1<j<*length*(M^k)$\}$
  {The Path-Loss model (Eq. 6) is used to compute distance when the RSSI is not found and there is no σ-closest entry in the map}
  if σ-closest = NULL **then**
    return pathLossRssiToDistance($R^{(k)}$)
  else
    if *length*(σ-closest) = 1 **then**
      d = σ-closest[0], r = σ-closest[1], w = σ-closest[2]
      accuracy = w/(w + N)
      return d
    else
      {Distance in the pair with the highest weight is returned}
      max_weight = $\{m_i = <(d_i, r_i)|w_i > / \ m_i \in$ σ-closest, w_i=max$\{w_j\}$, 1<j<*length*(σ-closest)$\}$
      if *length*(max_weight) = 1 **then**
        d = max_weight[0], r = max_weight[1], w = max_weight[2]
        accuracy = w/(w + N)
        return d
      else
        {At this state, the map has more than one closest entry to $R^{(k)}$ with different distances but with same weight. In the next iterations the weighting procedure will discriminate them. Meanwhile the log-distance path loss is used.}
        return pathLossRssiToDistance($R^{(k)}$)
      end if
    end if
  end if

$$accuracy = \frac{w}{w + N} \tag{5}$$

If the set of σ-closest entries contains more than one entry, the one with the highest weight is considered and the entry's distance is returned as the estimated distance. The accuracy of this estimation is determined as previously by Eq. 5. In all other cases, the distance is determined using the Path-Loss formula expressed in Eq. 6.

$$PL_{dB} = \overline{PL_0} + 10\eta\log_{10}(d/d_0) + X_\sigma \tag{6}$$

Where PL_{dB} is the path loss in *dB*, *d* the distance between the transmitter and the receiver in meter, η the path loss exponent that varies from 2 to 6 depending to the environment, d_0 the distance at the reference point (1 meter) and $\overline{PL_0}$ the path loss in *dB* at the reference point. Distances obtained from the mapping table are preferred to those from the Path-Loss model because they are based on real RSSI-distance samples. However, as it can take some time to obtain a significant number of accurate RSSI-distance samples, the Path-Loss model is also needed.

The distances obtained in this stage will be used for the cattle's collar localization process described in the following paragraphs.

C. Gateway algorithm for localization phase

In the localization phase, the Non-linear Least Square Fitting (NLSF) method is used to determine the position P_{B_i} of the Beacon-node with the estimated distances: $d^{RSSI}_{B_i,G_1}$, $d^{RSSI}_{B_i,G_2}$, $d^{RSSI}_{B_i,G_3}$ and $d^{RSSI}_{B_i,GW}$ between Beacon-node B_i and respectively GPS-nodes G_1, G_2, G_3, and Gateway *GW*. With NLSF, the localization issue is to determine for any given point X how well X can replace P_{B_i}. To do so, we calculate the distances between X and all anchor nodes (GPS-nodes and GW). If those distances perfectly match with $d^{RSSI}_{B_i,G_1}$, $d^{RSSI}_{B_i,G_2}$, $d^{RSSI}_{B_i,G_3}$ and $d^{RSSI}_{B_i,GW}$, then X is considered to be a good estimation of P_{B_i}. The more X deviates from these distances, the farther it is assumed to be from P_{B_i}. Therefore, finding the position of P_{B_i} is equivalent to minimizing the distance difference expressed by Eqs. 7, 8, 9, and 10.

$$e1 = d^{RSSI}_{B_i,G_1} - d_{X,G_1} \tag{7}$$

$$e2 = d^{RSSI}_{B_i,G_2} - d_{X,G_2} \tag{8}$$

$$e3 = d^{RSSI}_{B_i,G_3} - d_{X,G_3} \tag{9}$$

$$e4 = d^{RSSI}_{B_i,GW} - d_{X,GW} \tag{10}$$

The estimated position of P_{B_i} is point X that minimizes the Mean Squared Error (MSE), see Eq. 11. Practically, position X must be somehow initialized. In our experiment, we initialize X with the mean GPS coordinates computed on the coordinates of the GPS-nodes and the Gateway.

$$mse = \frac{\sum_{i=1}^{4} e_i^2}{4} \tag{11}$$

III. Experimentations and results

A. Preliminary tests

We first set up an experiment to evaluate the accuracy of distances obtained from GPS and distances from the Path-Loss model. The deployment consisted in a Gateway and an end-device with a GPS module in Line-of-Sight (LoS) condition. The end-device is moved successively from 5m to 100m from the gateway with an increment of 5m. 20 messages were sent at each location, each one providing the GPS coordinates of the end-device. This configuration design is motivated by the fact that animals in the herd are usually close together and so their collars are able to communicate on short distances in LoS condition. We also assumed that most collars are in near LoS condition with the Gateway because the animals usually move in an open rural area. GPS positions have been recorded as well as the corresponding RSSI values to perform distance estimation. The variation over distances of the collected RSSI are plotted in Fig. 4.

By nature the RSSI values fluctuate a lot because of their sensitivity to obstacles and multi-path effects. The RSSI we collected here are quite stable with a maximum standard deviation of about 3 (Fig. 4). In addition, we can observe in Fig. 4 that RSSI is quasi linear with regard to distance. Therefore the correlation between RSSI and distance is quite good and we can expect an accurate distance estimation through RSSI values especially with our real-time adjustment mechanism.

In Fig. 5 we can see the distance determined by the GPS coordinates. The distance estimation through the GPS coordinates is highly accurate (about 98%) with a very low Root Mean Square Error (RMSE) of approximately 0.77.

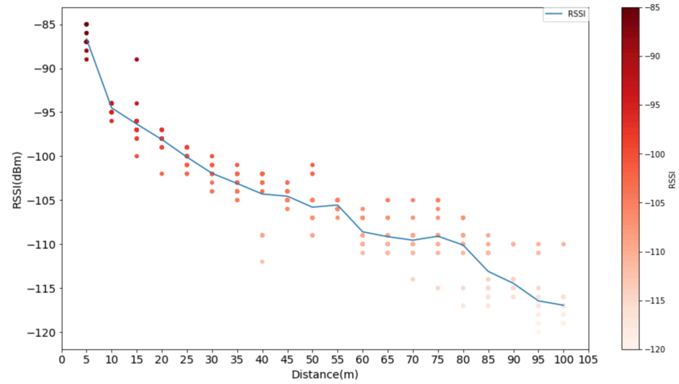

Fig. 4. RSSI values variations with regard to distances

The log-distance Path-Loss model is used as a rescue model when there is no correspondence in the RSSI-distance mapping table. Its accuracy has been evaluated with the collected data. The path loss at the reference point $\overline{PL_0}$ is determined in a preliminary test using the Friis model [11] and its value was found to be about 42dB. As the test is carried out in an area that can be considered as suburban, we choose 4.2 as path loss exponent. Observing the results, it appears that the path-loss model does not accurately estimate the distance especially for small distances as shown in Fig. 5. It can be seen in Fig. 5 that the accuracy of the model gets better when the distance goes higher. Therefore, the Path-Loss model although not as accurate as the RSSI mapping can be helpful for distances above 100m.

The continuous error is also tracked visualize the confidence area. For the distances estimated with the GPS coordinates, errors are almost non-existent and values are well in the confidence area. On the other hand, errors with log-distance estimation are more important but 70% of values are still in the confidence area. This is summarized in Fig. 5.

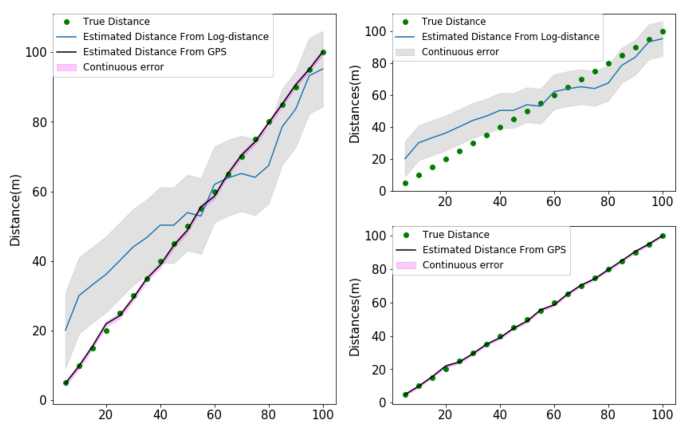

Fig. 5. Continuous errors for distance estimated with GPS and log-distance

B. In-situ deployment

We then evaluate our proposed dynamic and continuous RSSI-distance mapping mechanism with an in-situ deployment of 3 GPS-nodes (G_1, G_2 and G_3), two Beacon-nodes (B_1 and B_2) and one gateway (GW) as illustrated in Fig. 6. We summarized below the results obtained on the deployed network where the Path-Loss model is first used to determine distances when there are few samples of collected RSSI-distance pairs. The Beacon-nodes are represented by orange markers and their estimated position by blue markers. The results show that the accuracy of the Path-Loss model localization can greatly diverge from the real position, see Fig. 6(left).

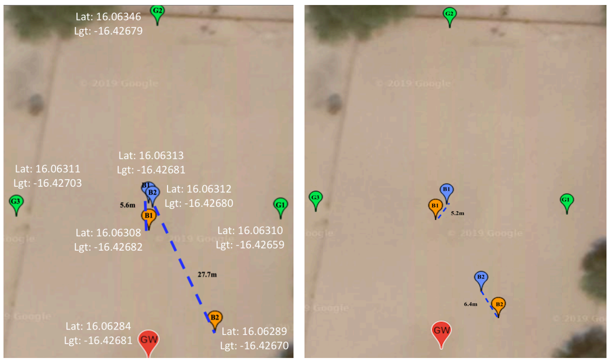

Fig. 6. Localization of two Beacon-nodes B_1 and B_2. (left) with Path-Loss model, (right) with RSSI-distance pairs obtained at run-time

After more RSSI-distance samples are collected, we can see in Fig. 6(right) that the estimated position of Beacon-nodes B_1 and B_2 are greatly improved. Here, as we have good LoS conditions, only a few number of true RSSI-distance pairs are sufficient to populate the mapping table in order to accurately determine the distances between the Beacon-nodes and the GPS-nodes acting as anchors, allowing for a much more accurate localization of those Beacon-Nodes.

C. Autonomy and Scalability

The collars are built from the low-cost and low-power IoT framework developed in the H2020 EU WAZIUP project which objective is to empower rural areas of African countries with low-cost yet efficient IoT technology. Using an Arduino ProMini (ATMega328P, 8MHz, 3.3v), a node draws about 40mA when transmitting (LoRa), 15mA when receiving and 5uA in sleep mode. With a beacon interval of 20mins and waking up for 3 GPS-nodes, a Beacon-node draws on an hour an average of $(3*(3*2s*15mA+1s*40mA)+(3600s-3*7s)*0.005mA)/3600s = 0.1133mA$, counting 2s for each packet reception window (including guard time) and 1s for the transmission of the beacon. With 2AA batteries (2500mA) the autonomy is well above 2 years.

For the GPS-node, a GPS module is added to the node. Power to the GPS module is provided through a small MOFSET transistor in order to completely power it down. In acquisition mode, the GPS consumes about 55mA (NEO 7M/M8N). The first GPS fix (when powering the collar for the first time) is usually obtained in about 30s. Each time the GPS is powered on again it can be considered as a cold start but a GPS fix can usually be obtained in about 4s to 6s. Not taking into account the first fix, an average fix time of 5s would give an average consumption of $(3*(5s*55mA+1s*40mA)+(3600s-3*6s)*0.005mA)/3600s = 0.2673mA$. The expected autonomy can be more than a year. By reducing the number of required GPS-node, the entire localization system can be easily deployed and maintained. Regarding scalability, a single gateway can easily handle more than a hundred collars, especially with Listen-Before-Talk and CSMA-like backoff.

IV. Acknowledgments

WAZIUP project has received funding from the EU H2020 research program under grant agreement No 687607. The authors also acknowledge the support of both CEA MITIC (The African Center of Excellence in Mathematics, Computer Science and ICT) and the IRD Sense-South GDRI.

V. Conclusions

In this paper, we proposed a dynamic and continuous RSSI-distance mapping mechanism using LoRa networks to localize cattle equipped with collars. The objective is to accurately localize collars without GPS and minimizing the number of collars with GPS. We proposed an original solution to improve the distance estimation scheme with adaptive RSSI-distance mapping algorithms that can refine the estimations at run-time by using messages exchanged between collars and the gateway. The advantage of the proposed approach is to seamlessly take into account the impact of the physical environment as true RSSI-distance mapping are continuously collected as animal move in the grazing area. Preliminary experimentation results with in-situ deployment showed the validity of the approach with accurate localization of non-GPS collars after a few rounds of true RSSI-distance mapping. Future work will test the dynamic RSSI-distance estimation approach in a larger variety of environments.

References

[1] W. Bakkali, M. Kieffer, M. Lalam and T. Lestable. Kalman filter-based localization for Internet of Things LoRaWAN end points. In *PIMRC*. IEEE, 2017.

[2] W. Choi, Y.-S. Chang, Y. Jung and J. Song. Low-power LoRa signal-based outdoor positioning using fingerprint algorithm. *ISPRS International Journal of Geo-Information*, 7(11):440, 2018.

[3] LoRa Alliance Strategy Committee. LoRaWAN geolocation whitepaper. 2018.

[4] J. Danebjer and V. Halldórsson. A hybrid approach to GPS-free geolocation over LoRa. 2018.

[5] O. Dieng, B. Diop, O. Thiare and C. Pham. A study on IoT solutions for preventing cattle rustling in African context, 2017.

[6] B. C. Fargas and M. N. Petersen. Gps-free geolocation using lora in low-power wans. In *2017 Global Internet of Things Summit (GIoTS)*. IEEE, 2017.

[7] Z. He, Y. Li, L. Pei and K. OKeefe. Enhanced gaussian process-based localization using a low power wide area network. *IEEE Communications Letters*, 23(1):164–167, 2019.

[8] K.-H. Lam, C.-C. Cheung and W.-C. Lee. New RSSI-based LoRa localization algorithms for very noisy outdoor environment. In *COMPSAC*, Vol. 2. IEEE, 2018.

[9] C. Pham. Investigating and experimenting CSMA channel access mechanisms for LoRa IoT networks. In *WCNC*. IEEE, 2018.

[10] N. Podevijn, D. Plets, J. Trogh, L. Martens, P. Suanet, K. Hendrikse and W. Joseph. TDoA-based outdoor positioning with tracking algorithm in a public LoRa network. *Wireless Communications and Mobile Computing*, 2018, 2018.

[11] T. S. Rappaport. Wireless communications–principles and practice. *Microwave Journal*, 45(12):128–129, 2002.