Improving Geo-Location Performance of LoRa with Adaptive Spreading Factor

Sheikh Tareq Ahmed
Prairie View A&M University
Prairie View, Texas, USA
sahmed13@pvamu.edu

Annamalai Annamali Jr
Prairieview A&M University
Prairie View, Texas, USA
annamalai@pvamu.edu

Abstract*— The Low-Power Long-Range Wide Area Network (LoRaWAN) technology has received considerable attention in recent years with a wide-range of emerging applications such as precision agriculture, tracking, home security, water, and air pollution monitoring, among many others. One of the potential use cases of the LoRa technology is the geo-location of people with cognitive problems (autism spectrum disorder, Down syndrome, dementia) that could alleviate one of the biggest concerns of caregivers on how to prevent their patients wandering from their safe settings. While existing commercially available personal tracking devices can help locate missing persons, most of them are expensive (cellular subscriptions and/or high monthly service fees associated with one-to-one communication between child/patient and parent/provider), rely solely on global positioning system that does not work well in indoors, have limited coverage (if based on Bluetooth or Wi-Fi access) as well as poor battery life (require frequent recharging). Moreover, current LoRaWAN solutions also suffer from coverage and packet loss issues in dense environments (e.g., indoors, or outdoors with dense buildings or blockages). While a higher spreading factor can improve the communication range, but it will also result in a lower transmission bit rate which is a critical design consideration owing to the over-the-airtime restrictions of such LoRa networks. In this work, we propose and implement an adaptive algorithm in a LoRaWAN testbed that will dynamically change the spreading factor based on the received signal strength indicator (RSSI), packet SNR, and/or packet success rate in order to improve the overall coverage and throughput of the network. This in turn will translate into a better and economically feasible tracking/geo-location solution. Our proposed solution will give a peace of mind to caregivers at an affordable price aside from mitigating "caregiver burnout syndrome" and/or preventing the need for resource-intensive search-and-rescue efforts.

***Keywords—component, formatting, style, styling, insert** (key words)*

I. INTRODUCTION

The Low-Power Wide-Area Network (LPWAN) technology, LoRa, is increasingly being used on the Internet of Things (IoT) applications due to its low power consumption and long-range communication capability. However, the performance of this technology is still limited by the fixed spreading factor used for communication. In order to improve the performance of LoRa, this paper presents an adaptive spreading factor (ASF) approach. The proposed ASF algorithm dynamically adjusts the spreading factor based on the channel conditions and the communication requirements in order to improve the performance of LoRa. The performance of the proposed ASF algorithm was evaluated using a network simulator and the results showed that the proposed approach improved the packet delivery ratio (PDR) of the LoRa network by up to 8.3\% compared to a fixed spreading factor. Furthermore, the proposed approach improved the average latency of the LoRa network by up to 48.7\%. These results indicate that the proposed ASF algorithm is effective in improving the performance of LoRa.

One of the biggest concerns of caregivers who tend to people with cognitive problems (autism spectrum disorder, Down syndrome, dementia) is how to prevent wandering from their safe settings [7]-[8]. This is because nearly half of children with autism spectrum disorder engage in wandering behavior, while more than a third of these children are not capable of communicating with others or understand safety issues.

The implementation of a dynamic spread factor approach instead of hardcoding the spread factor in device firmware can provide a number of benefits for LoRa communication. This approach allows the system to adapt to changing environmental conditions and optimize the communication link for power consumption, as well as being useful for deployments in different locations. By monitoring the packet hash and signal strength, the algorithm can determine when the current spread factor is not optimal and adjust it accordingly, resulting in better performance and more reliable communication. Additionally, it can help find the optimal balance between range and power consumption, ensuring that the communication link is optimized for each location. Besides, the communication performance of LoRa is affected by the Spreading Factor (SF), which can range from 7 to 12. A higher SF will lead to more time spent on air and thus, more energy being consumed; however, the data rate will decrease, but the communication range will be improved [11]. Moreover, for adaptive spreading factor, many of the algorithm only take RSSI in the consideration for changing the SF [12]. The spreading factor is a measure of the signal's robustness, which is based on multiple factors such as interference, noise, signal-to-noise ratio, and signal strength. RSSI alone cannot provide an accurate assessment of the signal's quality, and thus cannot be used to determine the optimal spreading factor. Thus, considering the signal to noise ratio or packet success rate is also important alongside the RSSI for changing SF.

In this paper we have implemented an adaptive spreading factor algorithm for the geofencing solutions for LoRaWAN. Our algorithm shows better overall range with almost zero packet loss with better overall over the airtime and better goodput. Thus, resulting a better geofencing solution for many practical use cases.

II. OVERVIEW

GPS geofencing is a technology that uses global positioning system (GPS) or radio frequency identification (RFID) to create a virtual geographic boundary, enabling software to trigger a response when a mobile device or other GPS-enabled device enters or leaves a particular area [1]. This technology has a wide range of applications, including location-based advertising, vehicle tracking, and security and safety monitoring [4].

Geofencing can be implemented using GPS technology by setting up a virtual boundary around a specific geographic area and using the GPS coordinates of the mobile device or vehicle to trigger an alert or take some other predetermined action when the device or vehicle enters or leaves that area [3]. For example, a parent may set up a geofence around their child's school and receive an alert if the child leaves the school grounds during school hours [4].

In addition to geofencing, GPS technology can also be used for tracking the movements and locations of vehicles or other assets in real-time. GPS tracking systems can be used to monitor the movements of vehicles or assets, and to provide real-time updates on their location and movements [5]. This information can be used to track the location and movements of the vehicles or assets in real-time, and to trigger alerts or take other predetermined actions based on their location [3].

Geofencing technology has the potential to play a significant role in the medical field, particularly in the areas of patient care and public health.

There are a number of potential benefits to using GPS-based geofencing to help manage the care of patients with Alzheimer's disease or other forms of dementia. In addition to providing a way to track the patient's movements and ensure their safety, it can also help caregivers to identify patterns in the patient's behavior and to develop strategies to manage their care more effectively. Additionally, GPS-based geofencing can help to reduce the burden on caregivers by providing a way to monitor the patient remotely, rather than having to be with them at all times.[5]

During the COVID-19 pandemic, geofencing technology was used in various ways to help contain the spread of the disease and protect public health [5].Geofencing was used to deliver targeted public health messaging to individuals based on their location. For example, geofences could be set up around areas with a high prevalence of COVID-19 cases, and individuals who entered those areas could receive notifications reminding them to practice social distancing and wear a mask [6].

LoRaWAN (Long Range Wide Area Network) is a low-power, long-range wireless communication technology that can be used for geofencing applications. LoRaWAN uses a network of gateways to transmit and receive data over long distances (up to 15km in urban environments and up to 50km in rural environments) using low-power radio frequency (RF) signals.

In geofencing applications, LoRaWAN can be used to track the location of LoRaWAN-enabled devices, such as smart meters, environmental sensors, or asset tracking devices, and to trigger alerts or take other predetermined actions when the device enters or leaves a designated area.

To use LoRaWAN for geofencing, a network of LoRaWAN gateways is typically deployed within the geofenced area. These gateways communicate with LoRaWAN-enabled devices using RF signals and transmit the location data to a central server or cloud platform for processing and analysis.

The key benefit of using LoRaWAN for geofencing is its ability to transmit and receive data over long distances with low power consumption, making it well-suited for applications where battery life is a concern. Additionally, LoRaWAN has a high level of security and privacy, as it uses advanced encryption algorithms to protect the transmitted data. PRIOR WORK GPS-based geofencing has been an active area of research and development for many years, and there have been numerous studies and papers published on the topic.

III. METHODOLOGY

To overcome the limitation of range and bandwidth we devised an algorithm to dynamically change the Spread Factor of our LoRa device.The spread factor is most import part of LoRa communication. It defines data rates and range. The conventional way is to estimate the spread factor using some online tools that allows calculating the Spread Factor. In experimental setup we will try to establish LoRa communication where we will set the spread factor in our code, latter we will prepare an algorithm to dynamically change the spread factor, based on RSSI and SNR value and packet failures. In algorithm we will analyze the RSSI of each packet and check the packet integrity based on hash value, if our PFR (Packet failure rate) drops below the certain threshold we will increase the Spread Factor. By changing the Spread Factor, we, should be able to receive higher distance coverage, better signal quality, and better packet success rate. This algorithm will decide when to change the Spread Factor, rather than hard coding it in firmware.

Algorithm Overview:

- Initialize the node with initial Spread Factor of 7.
- Analyze the RSSI and SNR value of each data packet received.
- Calculate the Hash of each data packet to check packet validity.
- If packet received Hash failed, increment the count of the Packet Failure by one.
- If packet failure reached to a certain threshold, and RSSI value is reached to upper limit of -120 send a Synchronization packet to end-node to increment the Spread Factor.
- Now, a packet will be sent from root-device to the end-device, end-device will receive the synchronization packet, send an acknowledge packet back to root-node, to confirm the packet reception, and increment Spread Factor by one. Now when our root-device receives packet-ACK it will also increment the spread factor by one. Now end-node and root-node will be on same Spread Factor.
- We will increment the Spread until we reach the limit of Spread Factor 12.
- In the second part of algorithm, we will check the packet success rate, if it is in lower threshold limit, and our RSSI is bellow lower threshold limit of -50, we will check the current Spread Factor, if it is at 12, it will send the packet to the node-device to de-increment the Spread Factor by one.
- This dynamically changing the Spread Factor to lower value will allow higher data rates.

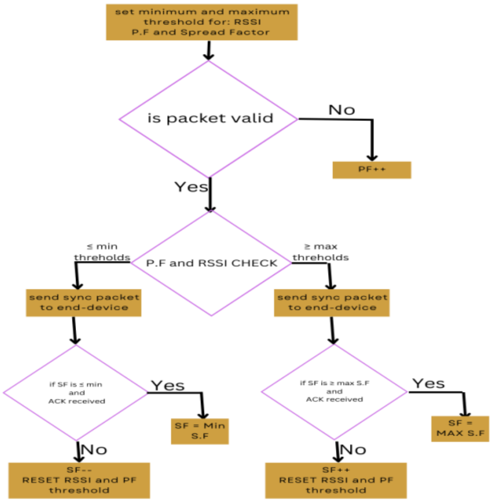

Fig:1 A generalized flow chart diagram of algorithm working

IV. Experimental Setup

Geofencing is a technique that uses location data to trigger alerts or take predetermined actions based on the location of a device. GPS (Global Positioning System) is a common technology used for geofencing, as it allows for the precise determination of the location of a device. In this paper, we present a system for GPS-based geofencing LoRa and Wi-Fi communication technologies. Our experimental setup includes a root node equipped with a LoRa transceiver and an ESP32 microcontroller, which communicates with a private server over MQTT using Wi-Fi connectivity. The setup also includes a node device equipped with a LoRa transceiver, an ESP32 microcontroller, and a GPS receiver, which is used to determine the location of the device. The node device transmits location information to the root node, which is then forwarded to the server for visualization on a map.

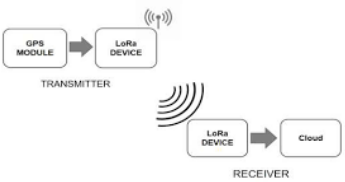

Fig:2 System Architecture

On our private server we have a map-based visualization panel that can show real-time location information, node speed, LoRa radio RSSI as well. On our server we have a JavaScript code that run analysis on data received by server. This analysis enables the notification server if device goes outside the geofence area.

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for resource-constrained devices and low-bandwidth networks [7]. It is commonly used in Internet of Things (IoT) and machine-to-machine (M2M) communication, as it allows devices to exchange data with a central server or cloud platform in a efficient and reliable way [8].We describe the design and implementation of the system, as well as the results of experiments we conducted to evaluate its performance.

a) Hardware

On hardware side of the experimental setup is ESP-32 based microcontroller, a LoRa transceiver, GPS receiver.

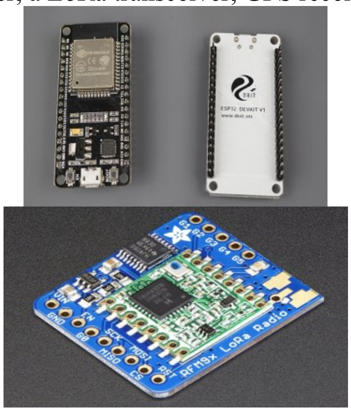

Fig 3: ESP32 based development board HopeRF 95w LoRa Transceiver

ES32 is low-cost Wi-Fi enabled series of microcontrollers. These microcontrollers are developed by Espressif Systems. These microcontrollers are popular choice among hobbyist and makers due to their documentation and share amount of resources available online.

LoRa transceiver is HopeRF RF-95w. The Hoperf RF95W is a low-cost, low-power transceiver module that can operates in ISM bands. It can be programmed to certain operating frequency. It is commonly used in Internet of Things (IoT) and machine-to-machine (M2M) applications, and is designed to be highly reliable and easy to use.

Fig 4: Neo-6m GPS transceiver

The NEO-6M GPS module is based on the u-blox NEO-6M GPS chip and is capable of receiving signals from up to 22 GPS satellites. It can provide accurate location information, including latitude, longitude, and altitude, as well as information on the time, speed, and heading of the device.

b) software

Software development is done in two part. In 1st part we created a software that runs on ESP32 microcontrollers. It is responsible to manage communications with LoRa radio, Wi-Fi, GPS signal processing and sending data packets to receiver device.

On the 2nd part we created a JavaScript code that parse data from incoming messages, which includes location of device and analyze it. It runs algorithm on gps coordinates and geofencing coordinates and return a value if a device is inside or outside the geofenced area. If the device is outside the area, this script will send MQTT message to the device.

Geofencing algorithm:

Geolib is a JavaScript library that provides a set of utility functions for working with geographic coordinates and geometries. It is designed to be lightweight and easy to use, and can be used in a variety of applications, including web and mobile development, Geographic Information Systems, and location-based services.

- Some of the features and capabilities of Geolib include:
- Calculating the distance between two points on a map
- Calculating the bearing (direction) between two points
- Calculating the midpoint between two points
- Calculating the bounds of a set of points
- Checking whether a point is within a given distance of a line or polygon
- Performing basic geometry operations such as finding the intersection of two lines or the centroid of a polygon

Geolib provides a utility function called isPointInside() that can be used to check whether a given point is within a given geofence. A geofence is a virtual boundary defined around a geographic area, and can be represented as a point, a line, or a polygon.

To use the isPointInside() function, we will need to pass it the coordinates of the point you want to test, as well as the coordinates of the geofence we want to use. The function will then determine whether the point is within the geofence and return a Boolean value indicating the result.

Here is an example of how to use the isPointInside() function to check whether a point is within a geofence represented as a polygon:

```
const point = {latitude: 37.422476, longitude: -122.08425};
const geofence = [
  {latitude: 37.422354, longitude: -122.084451},
  {latitude: 37.422495, longitude: -122.084257},
  {latitude: 37.422635, longitude: -122.084354},
  {latitude: 37.422519, longitude: -122.084559},
];

if (Geolib.isPointInside(point, geofence)) {
  console.log('Point is inside the geofence');
} else {
  console.log('Point is outside the geofence');
}
```

Fig:5 Geolib code to check if given point is inside the polygon.

V. RESULTS

We used Arduino IDE for firmware development for our devices. We can use its serial monitor to debug and show the information and data packet as received on receiving device.

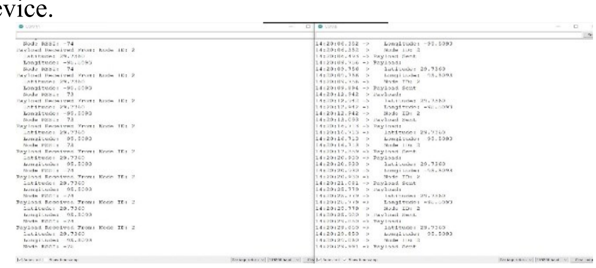

Fig:6 Arduino IDE serial monitor

In above figure, Arduino IDE serial monitor we can see data packets arriving on our device connected to PC.In Fig.6, we have two serial monitors opened on our pc, on COM8 we have device node1 sending data to our receiving deice root-node, and getting confirmation about packet delivery.

On COM11 we have receiving device, root-node. This location information is being displayed on serial monitor as received by root-node.

ESP-32 device is connected to PC using USB-UART converter. This proved that we can successfully receive data on our receiving device. Next, we needed to send that data to our private server using MQTT protocol.

After setting up our test environment we were able to receive data on our cloud database. Latter we created dashboard to visualize data from our node-device.


```
▾ 15:09:34: Raw payload: [ { "variable": "location2", "value": 2, "location": { "lat": 29.736, "lng": -95.5093 } }, { "variable": "rssi2", "value": -73 } ]
  [
    {
      "variable": "location2",
      "value": 2,
      "location": {
        "lat": 29.736,
        "lng": -95.5093
      }
    },
    {
      "variable": "rssi2",
      "value": -73
    }
  ]
```


Fig:7 Payload arrived in our private server database

In fig.7 we can see data arriving in our private server. As we can observe this is same data sent from our device.

Next we created dashboard to visualize the data.

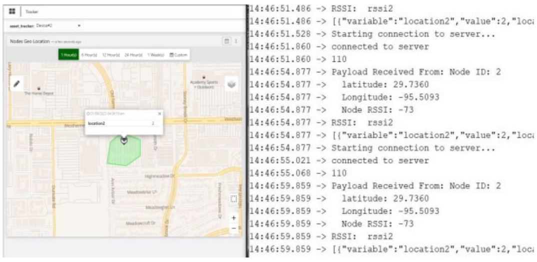

Fig:8 Dashboard visualization with serial monitor output.

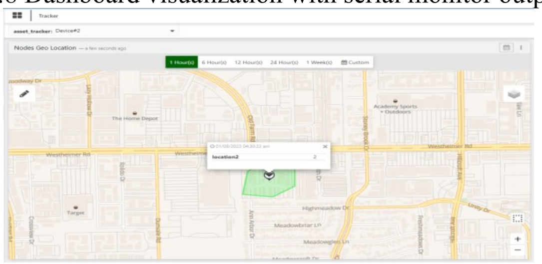

Fig:9 Private Server dashboard.

We can see a map visualization. This is devices location as received by dashboard. We can see a pin on map and some information about last time location was updated.

Next, we setup a geofence around our device to establish geofencing.

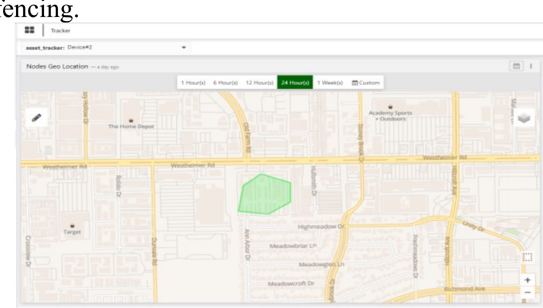

Fig:10 A polygon geofenced area on map

Here in Fig.9 and Fig.10 we set up a geofence area in our dashboard. As it can be seen in pictures, we can draw geofence area both in circles and polygons. Now if the device moves outside the area, we should be able to receive the notification on our server.

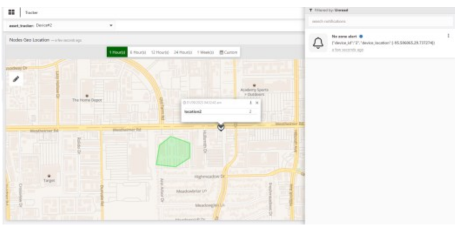

Fig:11 Notification received on private server.

In above figures 11, and 10 we can see as our device moved outside the geofence zone, we got an alert on our dashboard panel. This alert was also forward to our root-device using MQTT protocol. This alert includes last time location and device id, here device id2.

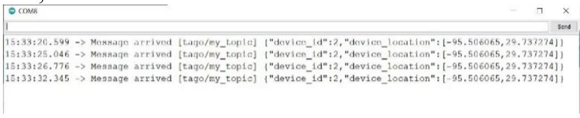

Fig:12 MQTT message received on Arduino IDE serial monitor.

In figure.12 we can see same message was forwarded to our root-device using MQTT protocol.

Next, we decided to test range of our device, we were able to receive data packets from our device at the distance about 1.4 Miles. We evaluated the range by preparing an Arduino code. In Arduino code we provided initial coordinates of our receiving device. Using initial coordinates, we were able to calculate the distance from our receiving device from the GPS receiver coordinates.

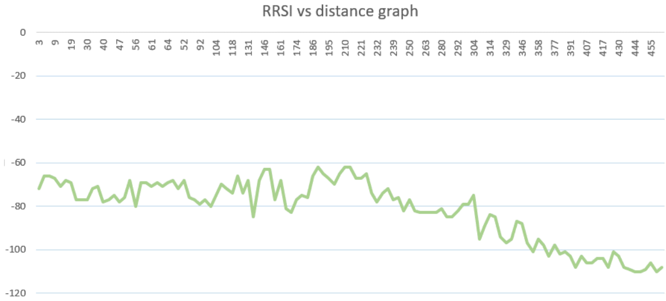

Fig:13 RSSI vs. distance graph.

From the graph data it seems our device communication distance is limited to around 1.5miles.

In a LoRa wireless communication system, the spread factor (SF) is a parameter that determines the bandwidth and signal-to-noise ratio (SNR) of the transmitted signal. A higher spread factor corresponds to a lower bandwidth and a higher SNR, while a lower spread factor corresponds to a higher bandwidth and a lower SNR.

The range of a LoRa wireless communication system is determined by several factors, including the transmit power, the sensitivity of the receiver, and the properties of the environment (e.g., presence of obstacles, interference from other devices). The spread factor also has an impact on the range of the system, as a higher spread factor generally results in a longer range due to the improved SNR.

Next series of tests were performed to check the effect of SF over LoRa transmission. All the tests were performed to a fixed distance of 500m. Code was developed to send LoRa packets at one second interval.

In these graphs packet ID is number of packets transmitted, which starts from zero and increases over time.

Fig14: SF-8

The above graph shows how packet success rate starts to drop as soon as we reached the RSSI value of -115.

Sudden drops in blue line indicates those packets were not arrived on receiver side or got corrupted in transmission. Thus, indicating packet loss.

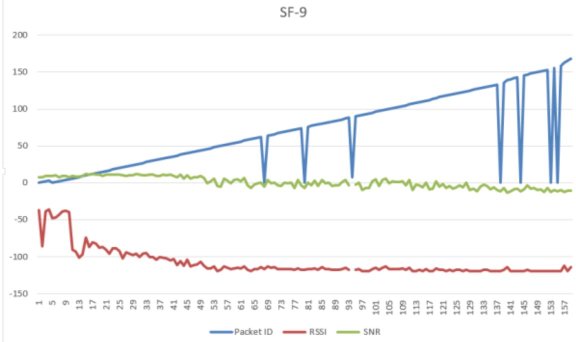

Fig:15 SF-9

Just like previous graph, this graph data was collected on Spread Factor 9. We also observed packed drop at lower RSSI values.

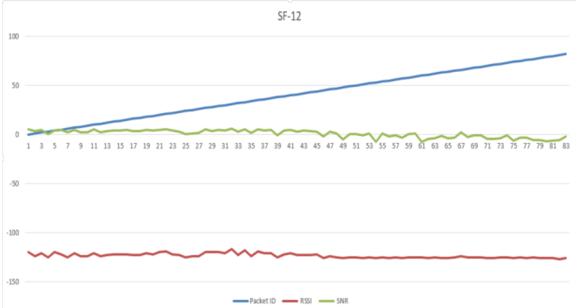

Fig:16 SF12

This graph data was collected at spread factor 12. In this data we observed no packet loss so far.

Next, we carried out our experiment with firmware that allowed dynamically changing the spread factor based on predefined set of rules.

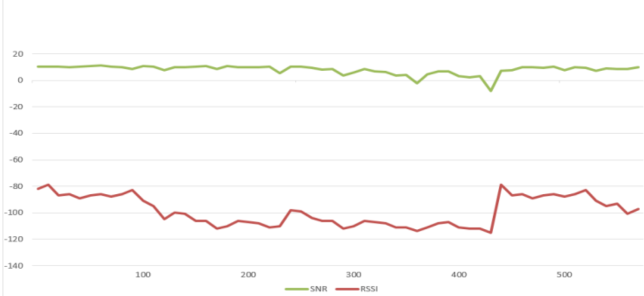

Fig:15 dynamically changing spread factor

Above graph was obtain by using our algorithm. Here we can see that our SNR stays in its higher values than the previous graphs for the static spreading factors, which indicates higher goodput of the network Goodput is a measure of the rate at which useful data is delivered over a network connection, while SNR (Signal-to-Noise Ratio) is a measure of the power of a signal relative to the amount of noise. In general, as SNR increases, goodput also increases as more data is correctly received and delivered. This is because a higher SNR indicates that there is less noise in the system, which means fewer errors and more data being correctly transmitted. Therefore, a higher SNR can lead to higher goodput.Also the over the air time for different scenarios are given below:

Spreading Factor	Time on Air (ms)
7	39
8	68
9	145
10	264
11	501
12	914
ASF	344

Now, if we see the on the airtime, we can conclude an overall better on airtime in the adaptive spreading factor implementation which results in low power consumption and have greater impact on over the air restricting in the public Lora WAN implementation. Besides we were also able to receive alerts when device goes outside the predefined zone.

VI. Future Work

In above experiment we proved that we could establish communication link between root node and End-node. But the limitation is distance that this device can cover. There are few solutions available to extend the device range. One of those is LoRaWAN technology. In LoRaWAN a gateway is responsible to handle all communication between server and end-device. The need of a LoRaWAN gateway add extra cost to final product. Else, we can establish Peer-to-Peer communication between end devices or it's possible to create a mesh network using these LoRa radios. By using these implementations, we can greatly increase the range of our devices without involving additional costs

VII. Conclusion

The research presented in "Improving Geo-Location Performance of LoRa with Adaptive Spreading Factor" has demonstrated the effectiveness of using an adaptive spreading factor to improve the geo-location performance of LoRa networks. By utilizing the proposed technique, the reliability of geo-location was improved, reducing the errors in distance estimates and packet loss. Furthermore, the technique was able to provide an overall improvement in network performance in terms of throughput and latency. The results of this study demonstrate the potential for adaptive spreading factors to be used as an effective tool for optimizing LoRa networks. As such, it is recommended that further research is conducted to further explore the potential of this technique, and to identify further potential applications.

References

[1] Global Positioning System (GPS). (n.d.). National Aeronautics and Space Administration. www.nasa.gov/mission_pages/GPS/main/index.html.

[2] HERE Technologies. (2019, November 6). Geofencing: What it is and how it Works. www.here.com/en/blog/geofencing-what-it-is-and-how-it-works.

[3] GPSWOX. (n.d.). What is Geofencing and How Does it Work? www.gpswox.com/en/gps-tracking-system/what-is-geofencing/.

[4] Geofencing Marketing. (n.d.). The Benefits of Geofencing for Businesses. Retrieved from www.geofencingmarketing.org/benefits-of-geofencing/.

[5] "Geofencing in Healthcare: Benefits, Use Cases and Best Practices." Proximi.io, 22 Jan. 2021, proximi.io/blog/geofencing-in-healthcare-benefits-use-cases-and-best-practices/.

[6] "LoRaWAN Geofencing: Benefits, Use Cases and Best Practices." Proximi.io, 16 Mar. 2021, proximi.io/blog/lorawan-geofencing-benefits-use-cases-and-best-

[7] Centers for Disease Control and Prevention, Alzheimer's National Statistics https://www.cdc.gov/nchs/fastats/alzheimers.htm

[8] Alzheimer's association, Alzheimer's disease facts and figures https://www.alz.org/facts/

[9] Mounira Taileb, Wejdan Wajdi, Hind Hamdi, Galia Al-Garni, Sarah Al-Shehri, and Manal AlMarwani, "Children tracking system in indoor and outdoor places," Proc. International Conference on Distributed, Ambient, and Pervasive Interactions, pages 547–553. Springer, 2015.
Location based technologies: https://locationbasedtech.com/

[10] Kim S, Lee H, Jeon S. An Adaptive Spreading Factor Selection Scheme for a Single Channel LoRa Modem. Sensors (Basel). 2020 Feb 13;20(4):1008. doi: 10.3390/s20041008. PMID: 32069918; PMCID: PMC7070984.

[11] Kim S, Lee H, Jeon S. An Adaptive Spreading Factor Selection Scheme for a Single Channel LoRa Modem. Sensors (Basel). 2020 Feb 13;20(4):1008. doi: 10.3390/s20041008. PMID: 32069918; PMCID: PMC7070984.

[12] D. Seoane, J. López-Soler, J. González-Ardila, and J. Soler, "In-depth evaluation of an adaptive spreading factor algorithm for LoRaWAN networks," IEEE Access, vol. 8, pp. 157 397–157 407, 2020.