TrunkNet: Elephant Herd Tracking IoT Application, Design Decisions

Samvrit Srinath
sasrinath@ucsd.edu
University of California, San Diego
La Jolla, California, USA

ABSTRACT

This report presents the design considerations, energy modeling, structural components, and implementation feasibility of an IoT-based elephant herd tracking application. The system aims to monitor elephant migration patterns in Southern India (where the author is originally from), using a network of infrared-equipped sensor nodes. Inspired by other short range sensors like the ones used in Ring doorbells, the nodes will detect and localize elephants based on heat signatures. The report then concludes the two best wireless technologies for this application, NB-IoT and LoRa, based on a tradeoff analysis of power consumption, data throughput, and cost efficiency. The final design is both cost-effective and energy-efficient, promising a robust solution for real-world wildlife monitoring.

1 INTRODUCTION & APPLICATION MOTIVATION

I am originally from Southern India, (in particular a city called Chennai), but when I go to visit my relatives who live in more remote parts of South India, I often hear stories about elephants wandering into villages and causing damage. I am also an Elephant enthusiast and have always been fascinated by these Elephants. This project is a way for me to combine my passion for elephants with real-world design considerations.

1.1 Background

Elephants are a keystone species in the Indian subcontinent, playing a crucial role in forest ecosystems. However, human-elephant conflicts are a significant issue in regions where human settlements overlap with elephant habitats. Monitoring elephant migration patterns is essential for wildlife conservation and to mitigate human-elephant conflicts, as well as to preserve the Quality of Life for both humans and elephants. I localize my study to strictly Southern India to serve as a *pilot* for a larger scale project that could be implemented in other parts of India and the world.

Figure 1 shows the current elephant migration patterns, especially concentrated in South India, which bolster the

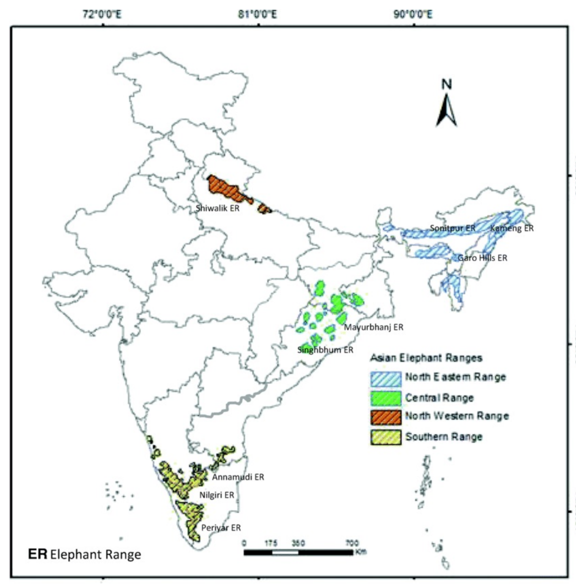

Figure 1: Map of Current Elephant Migration Patterns and concentrated Habitats in India. Notice the concentration of elephants in the *Annamadi*, *Nilgiri*, and *Periyar* Regions.

need for a system to monitor these elephants. The *Annamadi*, *Nilgiri*, and *Periyar* while located in remote areas, are proximal to many important cities like **Bangalore**, **Coimbatore**, and **Madurai**, aggregating to a total population of over 20 million people[12]. The need for a system to monitor these elephants is evident, and the system must be low-cost, low-power, and scalable to cover a large area.

Also another important design consideration is the fact that the elephants habitat are spread in a variety of terrains, usually in high elevation, dense forests, and remote areas. This provides some road blocks in terms of connectivity and power, and also makes it difficult for large scale infrastructure to be built (i.e. cell phone towers, power lines, etc.). So relative to each sensor node, the system must be self-sufficient in terms of power for our pilot study, and must be able to communicate over long distances.

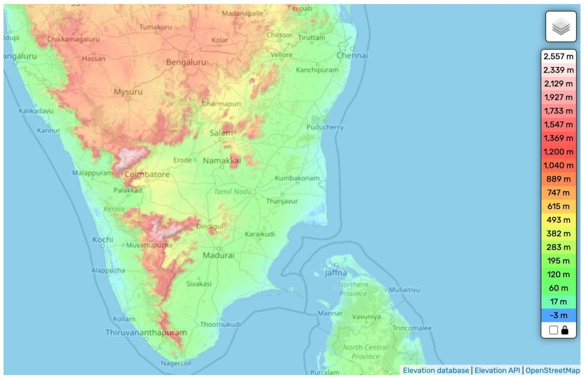

Figure 2: Ground Topology of the Elephant Habitat Area.

2 PART A: DEFINING THE APPLICATION

2.1 High-Level Description

Our goal is to implement a large-scale Elephant Tracking and Monitoring System across the **Neyyar Wildlife Sanctuary**. This Sanctuary is near cities like *Trivandrum* and *Madurai* and is encircled by many villages. For both the safety of elephants and of residents nearby to this sanctuary, it is crucial to montior the movements of these elephants. The system will consist of a set of **infrared-equipped** IoT nodes that will detect and beacon out their location based on heat signatures. Infrared will work exceptionally well as it is a **non-intrusive**, **non-visible** way to detect elephants. Subsequently, elephants also emit a lot of heat (as an innate property of body heat generated $\propto$ body mass), making it an ideal way to detect elephants.

The size of our deployment will be over several hundred square miles, making range and power efficiency imperative, coupled with the rough and forrested terrain of the sanctuary. There will be three main operating modes for the system:

(1) *Steady-State Sensing:* Periodic detection and transmission of location data.
(2) *Event-Driven Update:* Firmware updates and diagnostic data (e.g., when abnormal behavior is detected).
(3) *Emergency Mode:* In the event of an emergency, the system will switch to a high-power mode to ensure that the elephants are tracked and monitored. (i.e. always transmitting location data, sensing etc.)

The role of each of these is similar to what we have seen in our practical applications of IoT devices in Labs. Steady-State Sensing is akin to the Sleepy End Devices that we developed in Lab 3, where we periodically check in with the network and are awoken from sleep should our sensor detect anything or we need to listen from an update from the wider network. Event-Driven Update is akin to the firmware updates that we saw were a necessary condition in Homework 2. Finally, Emergency Mode is just the device will remain on and always transmitting data, in dire situations where an Elephant Habitat is moving towards a village, traffic, or other dangerous areas.

2.1.1 *Steady State Sensing.*

In this mode, the nodes will be in a low-power state, waking up periodically to check for elephants. Every 5 minutes, the nodes will simply periodically transmit their location data to a base station.

Our packet will look like:

- **Header:** 1 byte
- **Location Data(Latitude, Longitude):** 8 bytes
- **Timestamp:** 4 bytes
- **Elephant Detected/Present**: 1 byte
- **Battery Level:** 1 byte
- **Health Status:** 1 byte
- **Total:** 16 bytes

2.1.2 *Event-Driven Update.*

In this mode, the nodes will be doing diagnostic work, similar to the firmware updates we saw in Homework 2. Either by clearing out old data, or by checking for any abnormalities in the data that is being collected. Also firmware updates that will be carried out once every month will be done in this mode. (As our application is in a remote area, we cannot afford to have a lot of downtime for the system, so we will have to do firmware updates in a way that does not disrupt the system too much). **Assumption:** I'll assume that our firmware update will be of a similar scope to the ones we did in Homework 2, so a 10 MB update once a month will be done.

This means that we do this diagnostic clearing and firmware update once a month, batched, and the other times we are either in Steady State Sensing or Emergency Mode.

2.1.3 *Emergency Mode.*

In this mode, the nodes will be in a high-power state, always transmitting data to the base station. This will be done in the event of an emergency, where the elephants are moving towards a village, or a dangerous area. This will be done to ensure that the elephants are tracked and monitored at all times. This means that our sensor is always on, and always transmitting data (i.e we'll set a buffer of 10 seconds, and transmit data every 10 seconds). Rather than doing a periodic check and sleeping, this is more of an *active tracking mode*, where our Infrared Sensor is constantly "pinging the environment", and sending updates on movement of the elephants that are in immediate visibility.

In this mode, the packet structure remains the same to ease the transition between modes, and standardize the data parsing and data transmission efforts. However, the frequency of transmission is increased to every 10 seconds, and the data is always being transmitted.

- **Header:** 1 byte

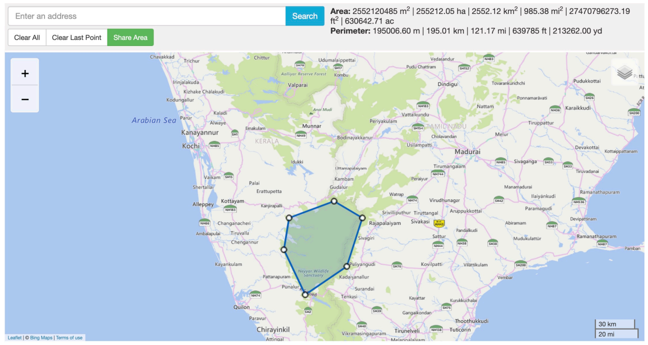

Figure 3: Neyyar Wildlife Sanctuary, Kerala, India. Total Area ≈ 985 mi^2.

- **Location Data(Latitude, Longitude):** 8 bytes
- **Timestamp:** 4 bytes
- **Elephant Detected/Present**: 1 byte
- **Battery Level:** 1 byte
- **Health Status:** 1 byte
- **Total:** 16 bytes

2.2 Data Workload

Simple dimensional analysis reveals the following data workload for our system. We use the formula

$$\text{Data Workload} = \frac{\text{Packet Size}}{\text{Time Interval (s)}} \times \frac{60 \text{ s}}{1 \text{ min}} \times \frac{60 \text{ min}}{1 \text{ hr}} \times \frac{24 \text{ hr}}{1 \text{ day}} \times \frac{30 \text{ days}}{1 \text{ month}}$$

With either Data Rate $= \frac{16 \text{ Bytes}}{5 \text{ min}}$ for Steady State Sensing, or Data Rate $= \frac{16 \text{ MB}}{10s}$ for Event-Driven Update.

Mode	Monthly Workload (MB)		Yearly Workload (MB)	
	Downlink	Uplink	Downlink	Uplink
Steady State Sensing	0.00	~0.13	0.00	~1.58
Event-Driven Update	10.00	Negligible	120.00	Negligible
Emergency Mode	0.00	~3.96	0.00	~47.46

Table 1: **Downlink and Uplink Data Workload per Mode (MB) - Monthly vs Yearly**

Table 1 shows the data workload for each mode, both monthly and yearly. Over our year-long pilot, each node will send roughly 1.58 MB in Steady State Sensing (if it remains in that mode for the entire year), receive 120 MB of downlink data in the form of firmware updates, and send 47.46 MB in Emergency Mode. Since many of our nodes will be in Steady State Sensing for a large part of the year, we can expect our Uplink Data Workload to be much closer to ~1.58 MB (perhaps there might be an aggregated usage of 1 month of Emergency Mode across the year, but this varies on the location of the node and the movement of herds). Roughly, we place an upper bound of ~ 47.46 MB for the Uplink Data Workload, in the worst case, if a node is proximal to a herd that is constantly moving towards a village or a dangerous area.

2.2.1 Speed. The speed of our data transmission is crucial, but the size of our packets are quite small. In Steady-State, we are transmitting 16 bytes every 5 minutes, which is a data rate of 0.0533 bytes per second. In Emergency Mode, we are transmitting 16 bytes every 10 seconds, which is a data rate of 1.6 bytes per second. Any Wireless Technology that we choose should be able to handle these data rates, which are quite small compared to the data rates that we see in our daily lives.

As a lower bound, we define that $> 20Kbps$ is the minimum data rate that we would need for our system. In the event of an emergency mode, we would need to be able to transmit data at a rate of $>$ 1.6 bytes per second. But higher data rates would enable us to send packets faster should we wanted to increase the frequency of our transmissions in a future iteration of our system.

2.3 Stakeholders

Organizations and Individuals that would be interested in the data collected by this system include:

- **Wildlife Conservation Agencies:** Agencies such as **WildlifeSOS**[17] and the **Asian Elephant Support**[15] would be interested in the data collected by this system. They would be interested in the movement of the elephants, and the health status of the elephants.

- **Municipalities and Local Communities**: Local Communities like Kattakada, Neyyattinkara, Ambasamudram, and Kadayan, which are all communities near the Neyyar Wildlife Sanctuary, would be interested in the data collected by this system. This could give insights as to where future elephant habitats could be, preventatively moving them away from villages and traffic, and protecting their constituents.
- **Forest Management Teams:** The Forest Management Teams in the Neyyar Wildlife Sanctuary would be interested in the data collected by this system. This could provide more info on where to do xeno-surveillance, and where to place more resources to protect the elephants.

2.4 Scale, Latency, and Reliability

Based on the aforementioned geographic area covered ~ 1000 sq.mi (2600 sq.km). we would assume a pilot of **1000 nodes**, with a potential to scale to 50000 nodes for even more granular data. The system should be able to provide near-real-time alerts (e.g., within minutes) in the event of an emergency, this is because due to the nature of our problem (elephants moving towards a village, or a dangerous area), we need to be able to act quickly. Occasional packet loss is acceptable, but overall system robustness is essential $< 5\%$ packet loss across the network. This metric is individually applied to each node, but a higher bar for "emergency nodes" that are in Emergency Mode. Emergency Nodes should ideally have $< 1\%$ packet loss, as they are the nodes that are in the most critical mode of operation, when elephants are proximal to a village or a dangerous area.

The reason why our pilot is so high is due to the range limitations of Infrared Sensors. PIR Sensors have a range of approximately 7 meters, but due to the animals that we are observing(elephants), that emit a lot of heat, we can assume that the sensitivity of our sensors will be much higher with respect to elephants versus humans. That being said, **1000 nodes**, gives each node a coverage area of 2600 meters per node, or around a 28 meter radius around each node. This is a reasonable coverage area, and we can expect that the elephants will be detected by at least one node in the network, or even a cluster of elephants will be easier to detect.

2.5 Deployment Constraints

The Nayyer Wildlife Sanctuary is a remote area, and as such, we have the following constraints on our deployment:

- Due to the mountainous and forested terrain of the sanctuary, we cannot rely on wired infrastructure. Our nodes will be battery-powered.
- It is difficult to put in larger pieces of infrastructure like towers or access points in the sanctuary, so either we must rely on large scale mesh networking, or we must rely on a technology that has a wide range.
- We can use already existing NB-IoT or Cellular Infrastructure based on my findings in Homework 2, but we must be able to have a system that can be maintained with minimal maintenance.
- The cost of each node should be in the order of \$50, with low recurring operational costs. This is because we are deploying a large number of nodes, and we must be able to do so in a cost-effective manner.

2.5.1 Labor and Maintenance. Our coverage area is quite wide, for the urposes of this report, we will assume that our technicians have access to an all-terrain vehicle to help them easily traverse the sanctuary (and is a reasonable assumption due to the ecological research initiatives already in place in the sanctuary). We will also assume that our technicians are well-versed in the technology that we are deploying, and can easily troubleshoot any issues that arise.

According to [1], the cost of hiring a Wireless Technician is 1913 rupees (\$22.25 according to 8.1) per month. The cost to also rent a Jeep to traverse the terrain for the day is 6500 INR [2] (\$77.99 according to 8.1). We will assume that our technicians can cover 25 nodes a day at an 8 hour work schedule, and we will hire 10 technicians to cover the 10000 square mile area.

This means, that it will take 4 days to deploy the entire network of sensors, as every day, all technicians will put up $\frac{25 \text{ nodes}}{1 technician} \times 20$ technicians $=$ 500 nodes. This is a reasonable time frame for deployment, and we can expect that the network will be up and running in a month.

The cost of labor and travel for the deployment is $(\frac{22.25}{\text{technician}} \times 20 \text{ technicians} + \frac{77.99}{\text{day}}) \times 20 \text{ days} = (\$522.99) \times 4 \text{ days} = \$2091.96$ for the deployment of the network. This is a reasonable cost for the deployment of the network, and we can expect that the network will be up and running in a month.

In terms of Maintanence, we will assume that the system will be able to operate for a full calendar year with minimal maintenance. This is a reasonable assumption, as we are deploying a large number of nodes, and we cannot afford to have a lot of downtime for the system. But in a 1 year maintence schedule, we will assume that technicians can *inspect* faster than they can install, so 50 nodes a day can be inspected by a technician. 10 Technicians will take on this effort of inspecting the nodes (just to minimize the impact/workload of exploring a remote area), and this will take 7 days to inspect all the nodes. The cost for this effort is $(\frac{22.25}{\text{technician}} \times 10 \text{ technicians} + \frac{77.99}{\text{day}}) \times 2 \text{ days} = (\$300.49) \times 2 \text{ days} = \$600.98$ for the inspection of the network. This is a reasonable cost for the inspection of the network, and can be done towards the end of our year long pilot.

In the grand scheme of things, this is not as expensive as what my initial estimates placeed the cost of the network at, but this is partly due in part to the fact that the salary wage for Technicians in India is much lower than in the United States (and is also quite oversaturated with a lot of technicians).

2.5.2 Additional Constraints. Weather should have minimal impact on the system. In particular, South India is notorious for having Monsoons and rain, so our system should be able to handle high winds and rain and this should minimally interfere with the communications of our system. Subsequently, under these conditions we can allow for a higher tolerance of packet loss ($< 10\%$) in the event of a monsoon or heavy rain, simply due to the fact that the weather is not conducive to the operation of the system. Also since we are in a forest, we must be able to handle the fact that the trees will absorb a lot of the signal that is being transmitted, which exacerbates the weather conditions.

Subsequently, our system should not use any bright or visual cues to detect the elephants, as this could potentially scare elephants away or cause them to act in a way that is not natural. This is why a constraint on the system is that it must be able to detect elephants using Infrared, as this is a non-intrusive and non-visible way to detect elephants.

3 WIRELESS TECHNOLOGIES RANKING

At a high level, some key features of our system are that:

- The system will be deployed in a remote location with no access to power for 1 year.
- The system will be required to transmit data over long distances.
- The system will be required to transmit very small amounts of data at regular intervals.
- The system will be required to operate for long periods of time without maintenance.
- The system will be required to operate in a harsh environment.

Given these requirements, we will now rank the wireless technologies ensembling on the following criteria:

- Power consumption
- Data throughput
- Cost efficiency
- Range

From worst to best, the ranking is as follows:

1. **Bluetooth**
2. **Wifi**
3. **802.15.4/Thread**
4. **Legacy Cellular(2G/3G)**
5. **High Performance Cellular(4G/5G)**
6. **LoRA**
7. **NB-IoT/LTE-M**

3.1 Bluetooth

Bluetooth is the least suitable technology for our elephant tracking application. While Bluetooth Low Energy (BLE) offers excellent power efficiency, its fundamental limitation is range—typically restricted to 10-100 meters in ideal conditions and significantly less in densely forested areas. The Neyyar Wildlife Sanctuary's 2600 sq.km area would require an impractically high density of nodes to create a functional network using Bluetooth.

Furthermore, Especially in the context of connections and advertising, Setting up Connections and also in Emergency Mode, where we need to have a connction that frequently transmit data, the Frequency Hopping Schematic of Bluetooth would be a hindrance and would adversely affect the power consumption of the device.

While we could set up our own GATT server and client, for nodes advertising and listening, our "gateways" that would be listening would not be able to support the 1000 nodes that we would need to cover the area. Also, nodes would be unable to connect to multiple Centrals or even connect to other sensor nodes to relay data. This would be a significant hindrance to our application, as a lack of peer-to-peer communication would mean that we would need to have a Central for every 100 meters, which would be infeasible. That being said, Concessions to be made are that Bluetooth is quite cost efficient and the technology is readily available. It is also quite easy to implement, and the data throughput is quite high. However, the range of Bluetooth is quite limited, and it would be difficult to cover the entire area of interest with Bluetooth.

3.2 Wifi

Wifi unfortunately is not a good choice for this application. 802.11 is quite power hungry and requires a lot of power to transmit data, especially if we're transmitting at the 2.4 GHz frequency. The range of Wifi is also quite limited, and it would be difficult to cover the entire area of interest with Wifi. It is infeasible to put APNs in the forest to cover the entire area of interest, as we do not have ready access to power, nor is the environment conducive for easy MAC (as at higher frequency channels, these can get lost or muddled in the terrain). While a Mesh Topology could be used, the power consumption of the meshes that would repeat the signal would be too high, and also the range of the mesh would also be limited, since our nodes have a domain of 260 meters to cover.

This innately violates our power consumption constraint, as the access points necessary to cover the area of interest would consume too much power. That being said, the data throughput of WiFi is quite high, and it is quite cost efficient to run. India has been making a push for lower cost WiFi, and it is quite cheap to run (Jio 5G/HighSpeed Wifi Push). However, the range of Wifi is quite limited, and it would be difficult to cover the entire area of interest with Wifi.

3.3 802.15.4/Thread

802.15.4/Thread is a decent choice for our application, however it is difficult to form a Mesh Topology in the terrain that our application is placed in. While as a WPAN, we could extract ~ 100 meters of range, the terrain would make it difficult for this range to be achieved, and likewise setting up gateways would be extremely difficult, due to a lack of readily available power. That being said, any device that is has a radio capable of FSK or QPSK would be able to communicate with our nodes, and the power consumption of the network is quite low. The data throughput of the network is also quite high, and it is quite cost efficient to run. However, the range of 802.15.4/Thread is quite limited, and it would be difficult to cover the entire area of interest with 802.15.4/Thread. We also gain redundancy due to the $4:1$ symbol rate, and the network is quite robust.

However, a glaring constrint is the mesh topology and the range is simply not sufficient enough to cover the area of interest. Seeing as we can't put up gateways, and the terrain would make it difficult to form a mesh, and would worsen the power consumption of our end nodes, there are other technologies that would be better suited for our application.

3.4 Legacy Cellular(2G/3G)

Legacy Cellular is quite over-engineered for our application. While 2G and 3G are quite power efficient, they are quite expensive to run, and the data throughput is quite high. As we saw in Homework 2, the cost for a 2G/3G connection is quite high (on the order of 7 per month), and while the data throughput is quite high, for our application at most we will be sending $1MB$ of data per second for our entire network, or 0.16 bytes per second per node (at least in Emergency Mode).

Another factor that exacerbates the 2G and 3G connection is the availability of such infrastructure. Since a property of our sensor nodes is that they require minimal maintanence, as India continues to push out for a 3G Sunset, the infrastructure for 3G will be dismantled, and it would be difficult to maintain the network. Jio, one of the largest telecom providers in India, has already started to dismantle their 3G network, and relying on such technology would be a poor choice[10]. What 2G and 3G has going for it is its topology, communicating to a cell tower over a long distance is exactly what this application requires (as we need to communicate over a long distance). However, the cost of maintaining such a network is quite high, and the power consumption of the network is also quite high. Also a Cellular Radio would be quite expensive to incorporate into our nodes, and would be quite power hungry as well (due to a lack of further optimizations seen in NB-IoT and 4G).

3.5 High Performance Cellular(4G/5G)

Our Application's ideal topology would be a star topology or at least a set of star topologies (tree) that communicate with one and another. High Performance Cellular, while extremely power hungry, has the range and existing infrastructure to readily deploy such a network. Homework 2 outlines the ease of deploying a 4G network of sensor nodes, which for 6 months would be on the order of \$20000 for 1000 nodes.

The data throughput of 4G and 5G is also quite high, but perhaps too high for our application. The data throughput of 4G and 5G is on the order of 100 Mbps, and we would only be sending $1MB$ of data per second for our entire network. That being said, at the 800 MhZ frequency, the range of 4G is quite high, and it would be easy to cover the entire area of interest with 4G. Jio and other India Cellular Carriers like Vodafone and Airtel have alraedy rolled out 4G support for the entirety of India, and our detection network could easily take advantage of this existing infrastructure. That being said, even with this existing infrastructure, the cost to operate a 4G network is quite high(and 5G adds more throughput that is unnecessary for our application and increases the power consumption of the network). While 4G seems to be an effective approach to our application, the cost to operate such a network could be quite higher, and a lack of enterprise solutions available for 4G sims could make it difficult to sustainably keep the network running for a long period of time > 1 year.

3.6 LoRA

LoRa is a great choice for this application, mainly because the Physical Layer of LoRa is quite power efficient, and has a much lower receive sensitivity. With the right antenna, we could easily cover the entire network with a few gateways along the perimeter of the Nayyer Wildlife Sanctuary. Rather than $100m$ in 802.15.4, Gateways could now be placed 10s of kilometers apart, and this would enable communication to occur across the entire network. With a TX Power of 20 dBm, and a RX Sensitivity of -119 dBm, the impact of brushes or trees would be less than those of other technologies like BLE or 802.15.4. And while LoRawans would work well in practice, the difficulty with Lora is Adoption.

Many of the existing Infrastructure in India is not LoRA enabled, and we would need to set up a translation gateway that would convert LoRA signals to 4G signals. Relative to the cost of the devices, as well as the installation cost of the gateways and the cost of the gateways themselves, the cost of the network would be quite low. The data throughput of LoRA is also quite high, and it is quite cost efficient to run. The Star Topology would also work well for most nodes, as every LoRa node that was within 20 km of a gateway would be able to communicate with the gateway[14]. The diameter of our provided area is around 50 km, so with on the order of 10 gateways, we could cover the entire area of interest. That being said, these gateways would also need a LoRa Network Server that links the LoRa Network to the already existing cellular infrastructure.

Thus, the only downfall of LoRA is the lack of Lora Infrastructure so far, unlike the United States with companies like Helium [7]. It should still be noted that LoRa is Operationally quite sound, due to its large range, low power consumption, and high data throughput, but the only thing that holds this technology back is adoption in India.

3.7 NB-IoT/LTE-M

Qualitatively, NB-IoT seems to be the best choice for our application. The power consumption of NB-IoT is quite low, as we can gate the power of our NB-IoT application to be a max of 20 dBm for TX, we can also allow for longer power off periods than conventional cellular devices, and optimize them for the IOT space. The data throughput of NB-IoT is also sufficient for our application, being 65 kbps and 26 kbps downlink,[13] but we have range comparable to LoRa (on the order of 10 kilometers) [3].

Power wise, NB-IoT is also much less power intensive than other cellular options, like $2G \sim 5G$, as we can enable power saving modes, limit TX Power, and also allow for "cellular" connections while being asleep for very long durations of time. The cost of NB-IoT is also much lower than paying for a 4G connection, due to enterprise pricing as well as the idea that our application uses very little data. In contrast to a finite $2.5Gb$ or $3Gb$ plan like Jio offers, NB-IoT plans would be on the order of a couple Megabyptes a month, so there would be no need for us as an application developer to pay for that much data that we would not use.

There is also 4G Coverage in that area, and we could easily set up a NB-IoT network that would communicate with the existing 4G infrastructure. The only downside of NB-IoT is the lack of infrastructure in India, and the lack of readily available NB-IoT modules. But many cellphone carries like Jio already offer NB-IoT support in various domains like fleet management and street lamps.

4 ENERGY MODELING

Consider that the two seemingly best performing LPWAN technologies, LoRa and NB-IoT, are possible choices for our Elephant Tracking Application. We will now compare the energy consumption of these two technologies.

As a constant, we will need to fix the Infrared Sensor that we will use in our application. For the purposes of this report, we will assume there is direct hardware compatability with the sensor and the LPWAN technology, and that we have done a control trial of using a smaller IR Sensor for detecting motion of the elephants at that specific range setting.

The IR sensor we will be using is the *GUMP's grocery HC-SR501 Infrared PIR Motion Sensor*[8]. This sensor has a range of 7 meters, and a power consumption of 60 μA, and since our operation will most likely be at the top end of its operational limit (farther to 7 meters), we will assume that the device is operating at $20V$. Roughly speaking, we use this sensor as it is small and inconspicuous, and can be placed in the forest without being noticed by the elephants. That being said, the efficacy of the sensor is still in question, especially in a dense forest environment.

4.1 LoRa

Our Application is based in Kerala + Tamil Nadu in India, so our LoRA specifications are different than the conventional US guidelines. In particular, in India we are able to use any Spreading Factor (SF) from SF7 to SF12, at 125 kHz bandwidth. This is because the Indian government has allocated the 865-867 MHz band for LoRaWAN use, which is different from the US guidelines. Another important thing is that according to [9], **there is no dwell or duty-cycle limitation on LoRA transmission**, according to Section 2.10.3 (IN865-867 Data Rate and End Device Output Power Encoding).

In the appendix, I've linked some of the LoRA quantities that are also found in screenshots (linked in appendix) from the Lora Guidelines. For our purposes, since we are using a Lora Module (I assume that we are using the SX1262 for transmit), and we will be comparing the power consumption of the device in both DC-DC and LDO mode. The LDO mode is more power hungry, and the DC-DC mode is more efficient.

Consider the operating modes that we defined earlier:

- Steady State Sensing
- Event-Driven Update
- Emergency Mode

4.1.1 Turning Off and On. Since our time interval is so large, turning off the radio and then turning on will be marginal relative to the power consumption of the device. This is because either we will sleep for a long time, or we will be receiving for a long time, and the power consumption relative to the major operations of the device is in the order of nanoAmps versus Milliamps (nearly a factor of 10^6). Subsequently, the datasheet provides the power consumption of the device in Sleep, but not necessarily the power consumption it takes to turn on the device → for the purposes of this report, we will assume that the power consumption of turning on the device is negligible (as a couple milliseconds relative to the 5 minute interval is not significant). In Steady State, the device will be in Sleep Mode for 5 minutes, wake up and second a transmission, and then go back to Sleep/standby. The power consumption for the Sleep duration is given by $1.2\mu A \times 3.3V * 300s = 1.188mJ$. This is with the crystal clock setting (XOSC) on. Then in Transmit this varies based on the power setting, let's see which mode is best for our application.

Now the possible configurations we have for LoRA are $SF7, SF8, SF9, SF10, SF11, SF12$ at $125kHz$ bandwidth. Then we can choose the power setting for the device, which can be $+14dBm$, $+17dBm$, $+20dBm$, $+22dBm$, as well as which mode we want to be in for hardware state (DC-DC or LDO).

For Steady State Sensing, we will sense for 1 second every 5 minutes, and then transmit accordingly based on our Spreading Factor prior to going back to standby.

In general we have the formula

$$E_{tx} = (\frac{P_{size}}{R} * I_{tx} * V_{opt}) * T_{tx} + E_{rest}$$

, where P_{size} is the packet size, R is the data rate, and the other terms are the current, the operating Voltage, and the standby Energy.

The Data rate will be dependent on the Spreading Factor, and the Current for Transmission will be dependent on the power setting.

4.2 Steady State Transmission

In essence, Table 4 contains simulated results of running the LoRa device under a Steady State configuration for 24 hours. The results are based on the power consumption of the device in different modes. Operating at SF7 with the lowest power setting of +14 dBm, the device consumes 30.310 J of energy over 24 hours, and this makes sense as the "chirps" are shorter and so messages can be transmitted faster. Intuitively, the higher the spreading factor, the more energy is consumed due to the longer chirps. At the highest end, SF12 with +22 dBm with a PA Target of +22 dBm, the device consumes 1.249 J of energy over 24 hours, nearly 30 times as much. This is a significant amount of energy, and it is clear that the device is not optimized for this setting. A configuration that strikes a sweet balance between energy consumption and data rate is SF7 with a +22 target with a Transmit Power of 22 dBm. This is the highest setting for the device, but by transmitting at a higher power, we still get a good data rate and a decent energy consumption of only 2.852 mJ over 24 hours, while still taking advantage of the LoRA's long range capabilities.

4.3 Event-Driven Update

In the cases where we do an update based on an event, ideally this is akin to a **Firmware Update** or a **Critical Event**. In this case, this is a downlink event where we will download a new patch, of 10 MB in size. Depending on our configuration, we can either use the DC-DC mode or the LDO mode. The DC-DC mode is more efficient, but the LDO mode is more power hungry.

It seems configuring the device to be in DC-DC mode is the best option for our firmware updates, as the power consumption is lower than in the LDO Mode. While we care about data integrity, the marginal optimization that LDO mode provides is not worth around 4 times the power consumption. Table **??** shows the power consumption of the device in the receive mode, and it is clear that the DC-DC mode is the best option for our application.

4.4 Emergency Mode

When the device is in Emergency Mode, we will be transmitting at the highest power setting, and we will be transmitting at the highest spreading factor. This is because we want to maximize the integrity of the device, and we want to ensure that the message is received. In this case, we will be transmitting at SF12, with a power setting of +22 dBm. This is the highest power setting for the device, and it is clear that the device is not optimized for this setting. The device consumes 1.638 J of energy over 24 hours, and this is a significant amount of energy. This is not a good setting for the device, and it is clear that the device is not optimized for this setting. We still aim to strike a good balance between power and data integrity, and we can still transmit at SF 12 with a +22 dBm power setting, but we will need to be more careful with our power consumption. A concern that is often brought up is that SF12 could cause an overlap in the chirps. This is a valid concern, but with our current packet structure of 16 bytes, we can still transmit at SF12 as the TX duration is $\frac{16*8}{250} = 0.512s$, which is less than the 10 second intermission between transmission.

4.5 NB-IoT

Our Application can rely on the already existing 4G Infrastructure that already covers the entirety of continental India. That being said, power wise we don't have to deal with the complexities of changing a Spreading Factor Unlike LoRA,

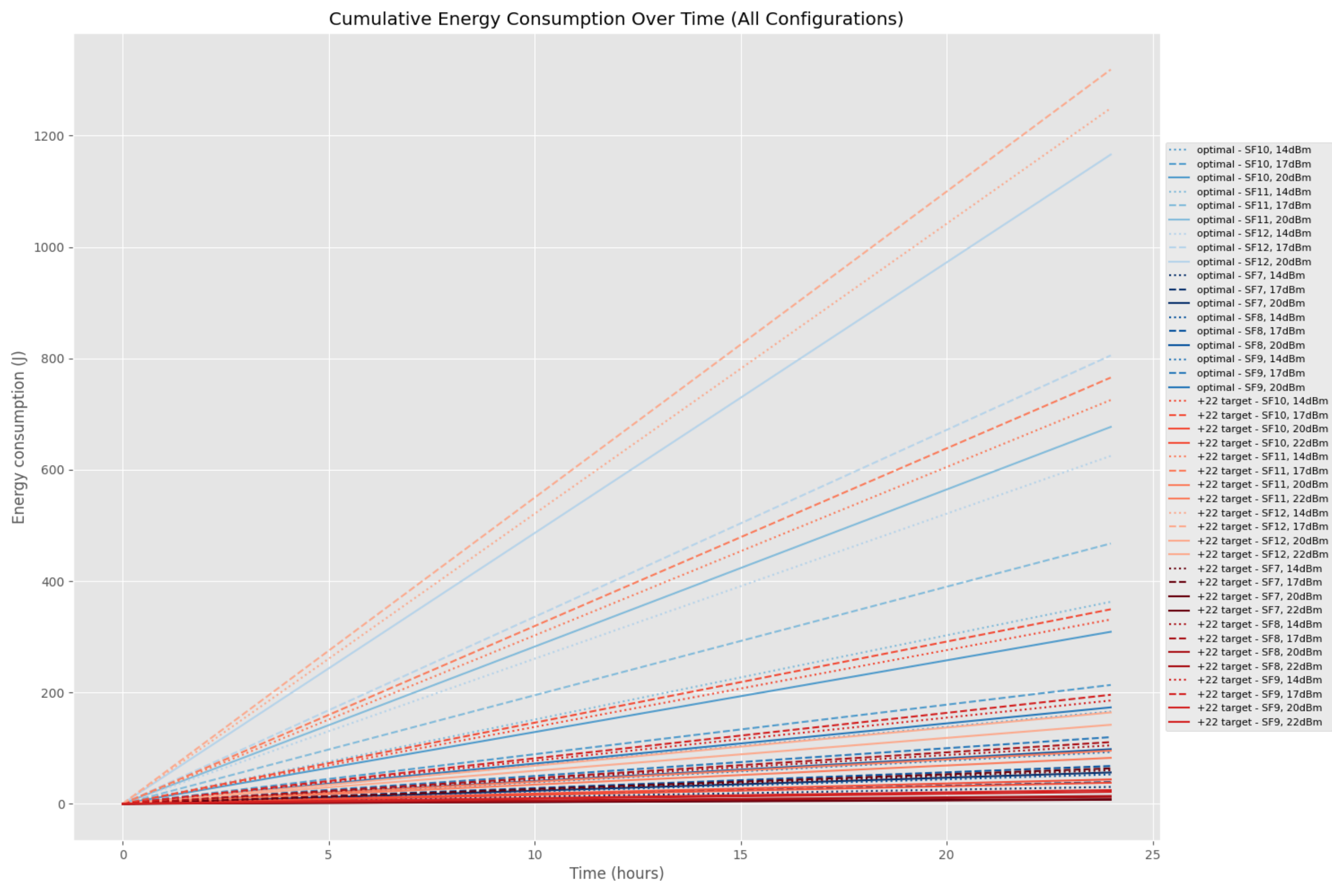

Figure 4: LoRa Device Power Consumption in Transmit Modes

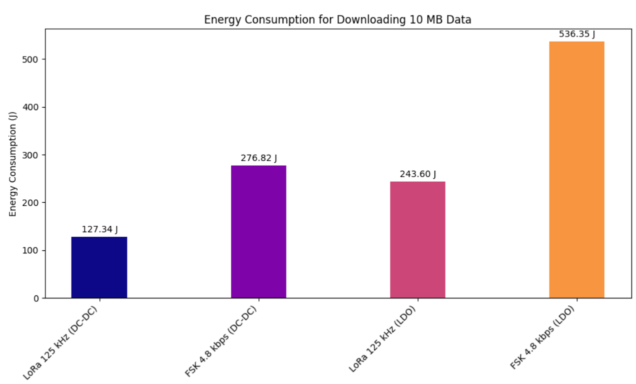

Figure 5: LoRa Device Power Consumption in Receiving Firmware Update

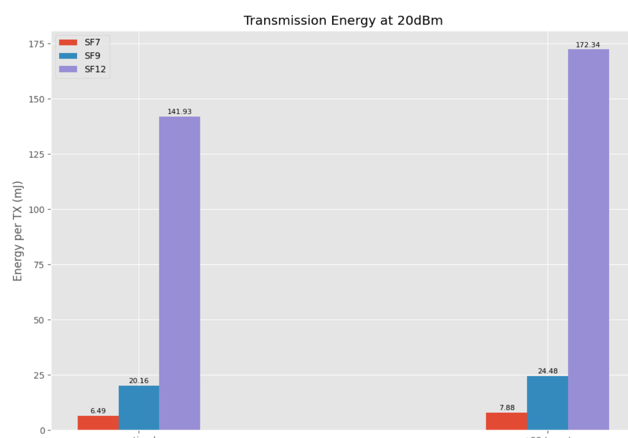

Figure 6: LoRa Device Power Consumption transmitting at +20 dBm

but we will have to tune some parameters to deal with our three different operating modes.

The NB-IoT device will be in a similar configuration to the LoRA device, but the power consumption will be different. The device will be in Standby Mode for 5 minutes, wake up and send a transmission, and then go back to Standby/standby.

The power consumption for this module is given by the same formula, but the current and voltage will be different. For brevity, **We will fix the module to the Sara R510S** for power consumption reasons (the initial deployment cost/cost of goods does not outweigh the module being performant for the entire year long pilot.)

Using the data sheet in [11], there are many configurations for which we can configure the chip. Due to Discontinuous Reception (DRX), and Extended Discontinuous Reception (eDRX), we can model the power consumption of the device using different intervals. This plays to the strength of NB-IOT, where sleeping for a long time is beneficial, as this means less wake ups to sync with the central tower, and less power consumption overall. Also, while the device is operating, the typical operating voltage is $3.8V$ but this value ranges from $3.3 \sim 4.4$ V.

The main tradeoff with the NB-IOT approach is that while sleep mode is around the same as LORA, transmission is very high (around $2 \sim 3$ the magnitude to transfer at the same power strength as LoRA). But this is is a high level artifact of the technology, that can only be absorobed in our implementation due to current adoption standards of LoRA in India (and the ubiquitous nature of 4G and 5G in the region).

4.6 Sleeping and Waking Up

The datasheet specified that Power Off takes 0.5 μA of current, but makes no mention of the power consumption of turning on the device. As with LoRA to keep a standardized metric across both wireless services, we will assume that the power consumption of turning on the device is negligible. This is because the use case of our application are such that the device will either be in standby for a long time (much larger than the time it takes to turn on the device), or the device will be transmitting for a long time (i.e. in Emergency Mode), or the device will be in a low power mode for a long time (i.e. in Steady State Sensing). Thus, the power consumption of turning on the device is negligible relative to the power consumption of the device in the major operations, especially when TX and RX are on the order of MilliAmps, and the difference from $0.5\mu A$ to a TX state is around $1000 times$ the power consumption (i.e. we would need to turn on many times to make the power consumption of turning on the device significant).

4.7 Steady State Sensing

As with LoRA, the formula we have still holds where

$$E_{tx} = (\frac{P_{size}}{R} * I_{tx} * V_{opt}) * T_{tx} + E_{rest}$$

, where P_{size} is the packet size, R is the data rate, and the other terms are the current, the operating Voltage, and the standby Energy. Using the narrowband standard $NB-02$ which is the standard for this SARA R500S product line(and since we are using NB-IOT to maximize range), our data rate is 140 kbps UL and 125 kbps DL. The current for transmission is dependent on the power setting, and the operating voltage is $3.8V$. Based on Figure 8, we can set an *Extended Disconnected Reception* window of nearly 65536 ms, which yields to an operating current of $1.1mA$. Thus, while either in standby we will have $1.1mA * 3.8V * 300s = 1.089J$ OR while off we'll have $0.5\mu A * 3.8V * 300s = 1.254mJ$. This is roughly the same as LoRA for Switching off the device, but the power while sleeping is quite a bit higher (1.5x).

What's key to understand though, is since the data rate is so high, the transmission time is much lower than LoRA, and the power consumption is much lower due to this. Consider the T_{tx} for a LoRA with SF12, $\frac{16*8}{250} = 0.512s$, but for NB-IOT, $\frac{16*8}{140000} = 0.1ms$. This is a signifcant difference and so even if the power consumption is higher, the Power it takes for the device to transmit is much lower due to the high data rate. **This observation holds for even larger packet sizes**, because as we increase $|P_{tx}|$, the duration for both LoRA and NB-IOT increases linearly, but the duration for NB-IOT is still much lower than LoRA.

This is where $NB-IOT$ has a leg up quantitatively on LoRA, as in spite of having higher power consumption for TX, the **Interval transpired for the device to transmit is much lower**. This is a key point to consider, as the device will be transmitting for a much shorter time, and thus the power consumption is much lower.

4.8 Event-Driven Update

In the cases where we do an update based on an event, ideally this is akin to a **Firmware Update** or a **Critical Event**. When we listen for a 10 MB update from the tower, let's assume that the tower is transmitting at $+23dBm$, then our RX current is now $195mA$, much higher than the LoRA's 10.1 mA when listening at 125 kHz Bandwidth. This is a significant difference, and it is clear that NB-IOT is not optimized for listening or frequent TX/RX Transfer. To listen in for our firmware Update once a month, this is roughly the same as LoRA, where it's on the order of a couple hundred Joules.

4.9 Emergency Mode

When the device is in Emergency Mode, we will be transmitting at the highest power setting, at +23 dBm. Based on our data so far, at this power setting, we use $395mA$ of energy per transmission, but since we're still transmitting at the same data rate, the transmission time is still much lower than LoRA. Roughly, at $395dBm$ when sleeping, we still only use $559.280\mu J$ of energy, and this is much lower than LoRA.

When standbying, we use $361.295mJ$ of energy, which is still an order of magnitude less than LoRA.

5 TAKEAWAYS AND BATTERIES

The optimal configuration for LoRA is to transmit at SF7 with a +22 dBm Power Setting, and this costs 78.662 mJ per day, and the optimal configuration for NB-IOT is to transmit at the highest power setting of +23 dBm, and this costs 559.280 μJ per day. This is a significant difference, and it is clear that NB-IOT is much more power efficient than LoRA. This is because the data rate for NB-IOT is much higher than LoRA, and the transmission time is much lower. This is a key point to consider, as the device will be transmitting for a much shorter time, and thus the power consumption is much lower.

6 PART C: HIGHER LEVEL FINDINGS + BATTERIES

Table 2: Comparison of LoRa and NB-IoT Features

Feature	LoRa	NB-IoT
Frequency	868 MHz	125 kHz
Sleep	1.2 µA	0.5 µA
RX	10.1 mA	195 mA
TX	118 mA	195 mA
UL Data Rate	5.47 kbps	140 kbps
DL Data Rate	5.47 kbps	125 kbps
V_{OPT}	3.3 V	3.8 V
E_{sleep}	1.188 mJ	0.570 mJ
E_{RX}	21.04 J	59.28 J
E_{TX}	9.11 mJ	1.34 mJ

Table 3: Comparison of LoRa and NB-IoT Features for Elephant Tracking Application – Energy

Ultimately where each technology shines is that for **receiving**, not at faster speeds but to conserve energy in general, **LoRA is the better choice.** This is because the power consumption of the device in receive mode is much lower than NB-IOT. However, for **transmitting**, **NB-IOT is the better choice.** This is because the power consumption of the device in transmit mode is much lower than LoRA. This is a key point to consider, as the device will be transmitting for a much shorter time, and thus the power consumption is much lower. That being said, the power consumption of the device in Sleep mode is slightly smaller for NB-IOT, but standby is a bit higher (0.8 to 1.1 mA). That being said for gateways and perhaps for nodes that might have to listen often, LoRA is the better choice. Arguably it comes down to the role of the device in our application, for nodes that frequently have to undergo patches or updates, LoRA will suit them better, as they can save more energy when listening to larger updates.

However, since the main proprietary goal of our application is to simply transmit frequently updates and not listen, NB-IOT is the better choice. This is because the power consumption of the device in transmit mode is much lower than LoRA, due to the fact that our transmit frequency is significantly higher ($140kbps$ vs $5.47kbps$). This is a key point to consider, as the device will be transmitting for a much shorter time, and thus the power consumption is much lower. That being said, if our NB-IOT service does not support such high speeds, metrics will lean more and more towards LoRA.

6.1 Battery

Consider a year long usage of the following application. This is modeled by the equation $E_{total} = E_{tx} + E_{rx} + E_{sleep}$, where E_{tx} is the energy consumed by the device in transmit mode, E_{rx} is the energy consumed by the device in receive mode, and E_{sleep} is the energy consumed by the device in sleep mode.

Every month, we do an E_{rx} for 10 MB, and then every 5 minutes, we send an update in Steady State sensing. For brevity, we generalize a trend in our network, s.t. **5% of our time we are in Emergency Mode, and the other 95% of the time in TX we are in Steady State Sensing**. This is a rough estimate, but it is a good approximation for our application. Intuitively, the search space is quite vast, and it's not practical to assume that more than 5% of the time we will be in Emergency Mode (that means there will be a rampant Emergency Threat for nearly 19 days under this model).

Thus, the total energy consumed by the device in a year is given by

$$E_{total} = 12 * E_{rx} + 12 * 24 * 30 * (0.95 * E_{steady} + 0.05 * E_{emerg})$$

. This is a rough estimate, but it is a good approximation for our application. It captures the essence of our end node behavior, that we will be updating once every month, and then transmitting every 5 minutes, and steady and emergency mode dictate how much we sleep and how often we transmit.

Thus, the total energy for LoRA is

$$\begin{aligned} E_{year} &= 12 * 243.605J + \\ 12 * 24 * 30 * &(0.95 * 56.046mJ + 0.05 * 1.638J) \\ &= 4090.91J \end{aligned}$$

And the total energy for NB-IOT is

$$\begin{aligned} E_{year} &= 12 * 497.28J + \\ 12 * 24 * 30 * &(0.95 * 361.295mJ + 0.05 * 559.280\mu J) = 6.5J \\ &= 9174.478J \end{aligned}$$

This is a significant difference, and it is clear that NB-IOT is much more power hungry than LoRA, that being said, existing infrastructure lends itself to a more optimal usage of NB-IOT, and the power consumption of the device in transmit mode is much lower than LoRA. Much of the change can also be chalked up to the fact that we chose a very unreliable medium of communication, choosing $SF7$, and $+20dBm$ in contrast to overthrottling the NB-IOT device to transmit at its max $+23dBm$ with the max power setting. While it may seem that the power consumption of the device in transmit mode is much higher than LoRA, balancing data integrity for only 2 times the power is a good tradeoff that I am willing to balance.

As such, a battery than can empirically support our system for a year must be a battery with at least $\frac{9175}{V_{nom}}$ mAh. Thus, we have two options:

- A Single 1.5 V D-Cell Battery with a capacity of around 15K mAh, some batteries are rated for $15K \sim 20K$ mAh, and this is a good choice for our application. This will last us for at least a year. At the standard $1.5V$ this is at least $22KmAh$, which could serve our end nodes for at least 2 years.[4]
- 3 AA Batteries in Series, with a capacity close to $2500mAh$ each, $7500 * 1.5V = 11250mAh$, which is a bit lower than the D-Cell, but still a good choice for our application. This will last us for at least a year.[5]

7 COMPARISON AND CONCLUSION

While our empirical understandings of both NB-IOT and LoRA and elephant migration are quite constrained, we have been able to make some general conclusions about the two technologies.

One thing that needs to be reiterated is the fact that the two technologies are in different places in terms of adoption, with NB-IoT being a more mature technology. This is reflected in the fact that NB-IoT has a higher data rate, lower latency, and better coverage than LoRa. However, LoRa has a much lower power consumption, which is a critical factor in our application.

Based on our extensive analysis of energy consumption for our Elephant Tracking Application, we conclude that NB-IoT outperforms LoRa on these critical metrics:

- **Data Speed:** NB-IoT has a higher data rate than LoRa, which is critical for our application.
- **Coverage:** NB-IoT has a better coverage than LoRa, which is critical for our application.
- **Ease of Implementation:** NB-IoT can leverage the already existing cellular infrastructure, which is a significant advantage over LoRa.

However, LoRa outperforms NB-IoT on the following critical metrics:

- **Power Consumption:** LoRa has a much lower power consumption than NB-IoT, which is critical for our application.
- **Cost:** LoRa has a much lower cost than NB-IoT, which is critical for our application.

NB-IoT Cellular modules are more expensive than LoRa modules, on the order of 50 per module [16]. That being said, the tradeoff here is that NB-IoT already has interworkings with the existing cellular infrastructure. Carriers such as Airtel, Vodafone, and Jio have already deployed large scale 4G networks in India, which LTE-M and NB-IoT can directly interface with. Clienting and contracting a large scale pilot, especially on the order of a 1000 devices would result in future deployment, less delay in the cost for our workers (quantified earlier in the paper), and a more robust system overall. The cost of the modules is a one-time cost, and while NB-IOT plans might have recurring payments, this might be cheaper then setting up our own LoRa network which has yet to mature. Unlike NB-IoT, LoRA reqiures external infrastructure like gateways, client and application servers to translate between the LoRa network and the internet. This is a significant disadvantage for LoRa, as it requires a lot of upfront investment in infrastructure.

Subsequently, in the case of massive emergency situations, the NB-IoT network can directly interface with other cellular mediums and can be used to send out emergency alerts should that become a future requirement of the system. Ensembled with a higher data rate, this could be a critical advantage in the future, especially with our 3rd Operating Mode **Emergency Mode**, which could interact directly with municipal services. While the Spreading Factor of LoRA can be adjusted to increase data rate, this would be inversely proportional to the range of the device, and also lead to a higher power consumption, reducing the lifetime of our end devices.

Empirically, NB-IoT also shines in constant reception. As indicated in 7, using different state schemes like LDO to preserve power, the energy consumption of NB-IoT is still lower than LoRa. 536.347 Joules for LoRA versus 497.28 Joules for NB-IoT. This is a significant advantage for NB-IoT, as the devices will be in a constant listening state, waiting for the next message from the elephants. If we were to use LoRa to scale up our devices into more of a mesh topology, where we were listening to multiple devices at once, the power consumption would be even greater as evidenced with our firmware update costs.

That being said, concessions for LoRA include its ability to be used in a mesh topology, which is a significant advantage for our application. This would allow us to have a more robust network, where the devices could communicate with each other, and relay messages to the base station. Emergency announcements would not have to be centralized to the base station, and could be relayed through meshes of devices instead. Redundancy in the network could be increased for very little cost, and the network could be more robust overall.

LoRA also allows for us to toggle between multiple different modulation schemes unlike NB-IoT, whether we want to increase throughput or preserve data integrity through higher spreading factors, we have the flexibility to do so as a property of the LoRA PHY ($SF7 \sim 12$). Likewise in India there is also no restriction on the dwell time for LoRA, which could be used intelligently to increase the range of our devices. And as a Standard, LoRA has a longer range than NB-IoT, which could be an advantage to us in the future, should LoRA mature and become more popular.

Qualitatively, LoRA shines when we want to send data in bursts and simply transmit messages without caring about packet integrity nor data rate. If power was not a concern and we could send data in bursts, LoRA would be the clear winner (i.e. aggregated statistics versus periodic monitoring), as LoRA has a much lower cost for transmission versus NB-IoT. IF our end devices become more "intelligent", and can aggregate data before sending it, transmission periods would shrink, and the power consumption of LoRA would be even less that what we have calculated earlier, as the device would be in a sleep state for longer periods of time, and the cost to send a message would be less. $\frac{byte}{cost_{LoRA}} << \frac{byte}{cost_{NB-IoT}}$.

Looking prospectively, LoRa serves as a valuable backup to NB-IoT, current work in this space primarily uses 4G networks to serve as their communication network, but applications like AI Tracking with elephants [6], could use a more "new" technology like LoRa as the backbone for any communication, especially in remote areas. When sending more granular data, LoRa could be more advantageous, as our current application sends a very rudimentary message (i.e. "Location + Is Present"), but larger packet sizes could lend themselves to LoRA.

Ultimately, the decision between NB-IoT and LoRa will depend on the specific requirements of the application. For our Elephant Tracking Application, we have decided to use NB-IoT as the primary communication technology, due to its higher data rate, better coverage, and ease of implementation, while accepting that 2 times the power is still supported by most common batteries, and does not drastically reduce the lifetime of our devices.

7.1 Application Requirements

At a high level our application will look like this:

- **Application Goal**: Monitor elephant migration patterns in Southern India.
- **Application Requirements**: Use an Infrared Sensor to detect and localize elephants based on heat signatures.
- **Application Modes**: Steady State Sensing, Event-Driven Update, Emergency Mode.
- **Application Data**: Location, Presence, Timestamp. Packet Size, 16 bytes.
- **Application Network**: NB-IoT.
- **Application Power Source**: D-Cell Battery or 3 AA Batteries.
- **Application Cost**: (54 per Cell Module + 8 per sensor + NB-IoT Subscription (roughly 7 dollars based on Cellular Prices) * 12) * 1000 + 2091.96 deployment fee (Labor and Travel). In aggregate 148, 091.66 for a 1000 device deployment for the first year.
- **Application Network Topology**: Star Topology.
- **Application Data Rate**: 140 kbps.
- **Application Range**: 10 km.

This paper has detailed the design considerations, energy modeling, and tradeoff analysis for an IoT-based elephant migration tracking system. By evaluating multiple wireless technologies, NB-IoT and LoRa emerge as the most viable solutions given our requirements. The final design is both cost-effective and energy-efficient, promising a robust solution for real-world wildlife monitoring.

Thanks for reading!

REFERENCES

[1] [n. d.]. . https://in.indeed.com/career/telecommunications-technician/salaries/Kochi--Kerala
[2] [n. d.]. . https://www.chennaitravels.in/mahindra-thar-rental
[3] [n. d.]. . https://docs.aws.amazon.com/whitepapers/latest/implementing-lpwan-solutions-with-aws/nb-iot.html
[4] [n. d.]. . https://uk.rs-online.com/web/content/discovery/ideas-and-advice/d-batteries-guide
[5] [n. d.]. . https://www.tycorun.com/blogs/news/a-complete-guide-to-aa-batteries
[6] [n. d.]. .
[7] Danish Khan. [n. d.]. *LoRaWan vs NBIoT: How Tata Comm and SenRa driving India's IoT space as telcos wait and watch.* https://telecom.economictimes.indiatimes.com/news/how-tata-comm-senra-driving-indias-iot-space-using-lorawan-as-telcos-still-finalise-nbiot-strategy/69983601
[8] Gump's Grocery. [n. d.]. . https://www.amazon.com/Gumps-grocery-HC-SR501-Infrared-Raspberry/dp/B08399XNTC/ref=asc_df_B08399XNTC?mcid=55b636e86eb53a79be3317cc4a7a68d9&hvocijid=7843168915725516339-B08399XNTC-&hvexpln=73&tag=hyprod-20&linkCode=df0&hvadid=721245378154&hvpos=&hvnetw=g&hvrand=7843168915725516339&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9031300&hvtargid=pla-2281435178098&psc=1
[9] LoRA. [n. d.]. https://lora-alliance.org/wp-content/uploads/2020/11/lorawan_regional_parameters_v1.0.3reva_0.pdf

[10] Live Mint. 2024. . https://www.livemint.com/industry/telecom/reliance-jio-urges-government-to-shut-down-2g-3g-services-in-the-country-vodafone-idea-heres-what-you-should-know-11706582831319.html
[11] NBIOT. [n. d.]. https://www.mouser.com/pdfDocs/SARA-R5_DataSheet_UBX-19016638.pdf
[12] Hitendra Padalia, Surajit Ghosh, Sudhakar Reddy Chintala, Subrata Nandy, Sarnam Singh, and Senthil Kumar. 2020. Assessment of historical forest cover loss and fragmentation in Asian elephant ranges in India. *Environmental Monitoring and Assessment* 191 (01 2020). https://doi.org/10.1007/s10661-019-7696-5
[13] Pat Pannuto. 2025. *Cellular*. https://drive.google.com/file/d/10yAwxdqM5ztwtH26nG6D6GQiUb0dbvnT/view
[14] Pat Pannuto. 2025. *LPWANs*. https://drive.google.com/file/d/111VJZlof3eLgUXOWDz4TBXm4581yQpT8/view Slide 24,87.
[15] The Asian Elephant Conservation Program. [n. d.]. . https://www.arocha.in/conservation/asian-elephant-conservation-program-aecp/
[16] SARA. [n. d.]. . https://www.u-blox.com/en/product/sara-r5-series
[17] Wildlife SOS. [n. d.]. . https://wildlifesos.org/

8 APPENDIX

8.1 INR Conversion Rate

The conversion rate today for USD to INR is 85.99 : 1

8.2 LoRa Energy Tables

8.3 NB-IoT Energy Tables

Spreading Factor	Transmission Power	PA Target	Joules (24 hours)
7	20	optimal	56.046 mJ
7	17	optimal	38.751 mJ
7	14	optimal	30.104 mJ
8	20	optimal	97.804 mJ
8	17	optimal	67.584 mJ
8	14	optimal	52.474 mJ
9	20	optimal	172.981 mJ
9	17	optimal	119.492 mJ
9	14	optimal	92.747 mJ
10	20	optimal	308.750 mJ
10	17	optimal	213.237 mJ
10	14	optimal	165.480 mJ
11	20	optimal	676.736 mJ
11	17	optimal	467.321 mJ
11	14	optimal	362.614 mJ
12	20	optimal	1.166 J
12	17	optimal	805.108 mJ
12	14	optimal	624.689 mJ
7	22	+22 target	78.662 mJ
7	20	+22 target	68.019 mJ
7	17	+22 target	63.363 mJ
7	14	+22 target	60.037 mJ
8	22	+22 target	137.323 mJ
8	20	+22 target	118.726 mJ
8	17	+22 target	110.590 mJ
8	14	+22 target	104.778 mJ
9	22	+22 target	242.929 mJ
9	20	+22 target	210.012 mJ
9	17	+22 target	195.611 mJ
9	14	+22 target	185.325 mJ
10	22	+22 target	433.652 mJ
10	20	+22 target	374.875 mJ
10	17	+22 target	349.160 mJ
10	14	+22 target	330.792 mJ
11	22	+22 target	950.585 mJ
11	20	+22 target	821.715 mJ
11	17	+22 target	765.334 mJ
11	14	+22 target	725.062 mJ
12	22	+22 target	1.638 J
12	20	+22 target	1.416 J
12	17	+22 target	1.319 J
12	14	+22 target	1.249 J

Table 4: **Energy Consumption Over 24 Hours for LoRA Steady State**

Table 5: Energy Consumption for Downloading 10 MB Data

Mode	Regulator	Energy (Joules)
LoRa 125 kHz	DC-DC	127.339 J
FSK 4.8 kbps	DC-DC	276.824 J
LoRa 125 kHz	LDO	243.605 J
FSK 4.8 kbps	LDO	536.347 J

Table 6: **LoRA Energy Consumption for Downloading 10 MB Data**

RX Power	Joules (24 hr)
0 dBm	255.01 J
8 dBm	293.27 J
14 dBm	357.02 J
20 dBm	433.52 J
23 dBm	497.28 J

Table 7: NBIOT Energy Consumption for Listening to 10 MB

8.4 LoRA and NbIOT Specs

TX Power	PA Target	Joules (24 hr)
0	Turn Off	264.105 μJ
0	Idle Sleep	361.000 mJ
8	Turn Off	279.114 μJ
8	Idle Sleep	361.015 mJ
14	Turn Off	304.129 μJ
14	Idle Sleep	361.040 mJ
20	Turn Off	334.147 μJ
20	Idle Sleep	361.070 mJ
23	Turn Off	359.161 μJ
23	Idle Sleep	361.095 mJ
23(max)	Turn Off	559.280 μJ
23(max)	Idle Sleep	361.295 mJ

Table 8: **Energy Consumption Over 24 Hours for NB-IoT**

DataRate	Configuration	Indicative physical bit rate [bit/s]
0	LoRa: SF12 / 125 kHz	250
1	LoRa: SF11 / 125 kHz	440
2	LoRa: SF10 / 125 kHz	980
3	LoRa: SF9 / 125 kHz	1760
4	LoRa: SF8 / 125 kHz	3125
5	LoRa: SF7 / 125 kHz	5470
6	RFU	RFU
7	FSK: 50 kbps	50000
8..15	RFU	

Table 70: IN865-867 TX Data rate table

Figure 7: Data Rate based on Spreading Factor for LoRA, INR Standards

Mode	Receive Mode (RX)	
	Condition	Power (mA)
Sleep Mode	Configuration retained	600 nA
Sleep Mode	Configuration retained + RC64k	1.2 μA
Standby Mode	RC13M, XOSC OFF	0.6 mA
Standby Mode	XOSC ON	0.8 mA
(ReceiveDC-DC Mode)	LoRa 125 kHz	4.6 mA
(Transmit PA Match)	Rx Boosted, FSK 4.8 kb/s	4.8 mA
	Rx Boosted, LoRa 125 kHz	5.3 mA
LoRa 868/915 MHz	FSK 4.8 kb/s	8 mA
(Receive LDO Mode)	LoRa 125 kHz	8.8 mA
(Transmit PA Match $+22dBm$)	Rx Boosted, FSK 4.8 kb/s	9.3 mA
	Rx Boosted, LoRa 125 kHz	10.1 mA

Table 9: **LoRa Device Power Consumption in Receive and Transmit Modes**

4.2.3 Current consumption

Mode	Condition	Tx power	Module	Min	Typ[11]	Max	Unit
Power-off mode	Average current value (power-off mode)	--	SARA-R510S		0.5		µA
			SARA-R500S SARA-R510M8S		62		µA
PSM deep-sleep mode	Average current value (PSM deep-sleep mode)	--	SARA-R510S		0.5		µA
			SARA-R500S SARA-R510M8S		62		µA
Cyclic idle / active mode (+UPSV=1)	Average current value (rock bottom)	--	All		1.0		mA
	Average current value (DRX = 2.56 s, PTW = 20.48 s, eDRX = 655.36 s)	--	All		1.1		mA
	Average current value (DRX = 2.56 s, PTW = 20.48 s, eDRX = 81.92 s)	--	All		1.2		mA
	Average current value (DRX = 2.56 s, PTW = 5.12 s, eDRX = 20.48 s)	--	All		1.2		mA
	Average current value (DRX = 2.56 s, no eDRX)	--	All		1.5		mA
	Average current value (DRX = 1.28 s, no eDRX)	--	All		2.0		mA
Active mode (+UPSV=0)	Average current value (DRX = 1.28 s)	--	All		25		mA
LTE Cat M1 connected mode	Average current value (Tx / Rx data transfer)	Minimum (–50 dBm)	All		95		mA
		0 dBm	All		100		mA
		8 dBm	All		115		mA
		14 dBm	All		140		mA
		20 dBm	All		170		mA
		Maximum (23 dBm)	All		195		mA
	Maximum current value (During Tx only)	Maximum (23 dBm)	All		395		mA

Table 11: VCC current consumption of SARA-R5 series modules with GNSS off

Figure 8: SARA Narrow-band IoT Module Power Consumption

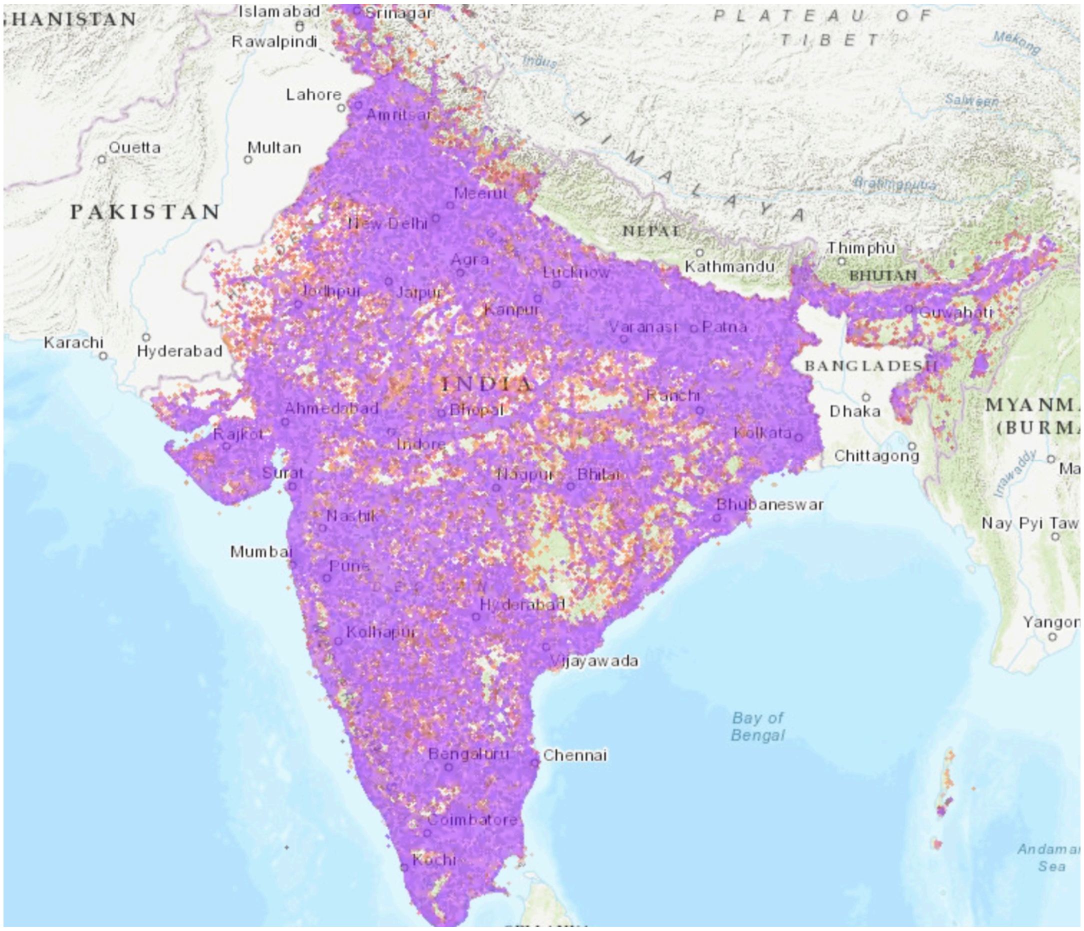

Figure 9: Coverage Comparison of 4G/NB-IoT in India via Jio