Precision agriculture in vineyards using a low-power wide-area wireless sensor network

Bruno Casu

Content

- LoRaWAN Server testing
- Review on the LoRa ADR process
- NUCLEO LoRa Testbench and extending the LoRaWAN Duty Cycle
- A Model for evaluating the energy cost in the End Nodes when transmitting a data frame
- Outlook

LoRaWAN Server testing

- The TTN LoRaWAN Stack was installed on the VM provided. The DataBase was created successfully and the server was started on http://localhost:1885.

```
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS                    PORTS                        NAMES
00497a5a0675   postgres:14    "docker-entrypoint.s…"   About an hour ago    Up About an hour          127.0.0.1:5432->5432/tcp     lorawan-stack-dev_postgres_1
0724ce02bb4c   redis:7        "docker-entrypoint.s…"   About an hour ago    Up About an hour          127.0.0.1:6379->6379/tcp     lorawan-stack-dev_redis_1
b664ced0a2fe   hello-world    "/hello"                 8 hours ago          Exited (0) 8 hours ago                                 compassionate_jemison
ubuntu@vecchio-vm-3:~/go/lorawan-stack$
```

```
ubuntu@vecchio-vm-3:~/go/lorawan-stack$ go run ./cmd/ttn-lw-stack/ -c ./config/stack//ttn-lw-stack.yml start
INFO    Setting up core component
WARN    No cookie hash key configured, generated a random one
WARN    No cookie block key configured, generated a random one
INFO    Setting up Identity Server
INFO    Setting up Gateway Server
INFO    Setting up Network Server
INFO    Setting up Application Server
INFO    Setting up Join Server
INFO    Setting up Console
INFO    Setting up Gateway Configuration Server
INFO    Setting up Device Template Converter
INFO    Setting up QR Code Generator
INFO    Setting up Packet Broker Agent
INFO    Setting up Device Repository
INFO    Setting up Device Claiming Server
INFO    Starting...
WARN    No cluster key configured, generated a random one       {"key": "51052aac84458469a0b1f64e5c3f81083d0fad875d57267f6cebf19a1201a81d", "namespace": "cluster"}
INFO    Listening for connections       {"address": ":1884", "namespace": "grpc", "protocol": "gRPC"}
INFO    Listening for connections       {"address": ":8884", "namespace": "grpc", "protocol": "gRPC/tls"}
INFO    Listening for connections       {"address": ":1885", "namespace": "web", "protocol": "Web"}
INFO    Listening for connections       {"address": ":8885", "namespace": "web", "protocol": "Web/tls"}
INFO    Listening for connections       {"address": ":1886", "namespace": "interop", "protocol": "Interop"}
INFO    Listening for connections       {"address": ":8886", "namespace": "interop", "protocol": "Interop/tls"}
```

LoRaWAN Server testing

- The Things Stack requires two configuration files when installing with Docker: `docker-compose.yml` and `ttn-lw-stack-docker.yml`.
- For production deployments the example server address `thethings.example.com` in `ttn-lw-stack-docker.yml` must be replaced (also additional settings must be changed).
- In the docker-compose.yml there is a call for the configuration file:

```
stack:
  image: thethingsnetwork/lorawan-stack
  entrypoint: ttn-lw-stack -c /config/ttn-lw-stack-docker.yml
  command: start
  restart: unless-stopped
```

LoRaWAN Server testing

An Example of **ttn-lw-stack-docker.yml** file can be found in the TTN documentation

```
# If Gateway Server enabled, defaults for "thethings.example.com":
gs:
  mqtt:
    public-address: "thethings.example.com:1882"
    public-tls-address: "thethings.example.com:8882"
  mqtt-v2:
    public-address: "thethings.example.com:1881"
    public-tls-address: "thethings.example.com:8881"

# If Gateway Configuration Server enabled, defaults for "thethings.example.com":
gcs:
  basic-station:
    default:
      lns-uri: "wss://thethings.example.com:8887"
  the-things-gateway:
    default:
      mqtt-server: "mqtts://thethings.example.com:8881"
```

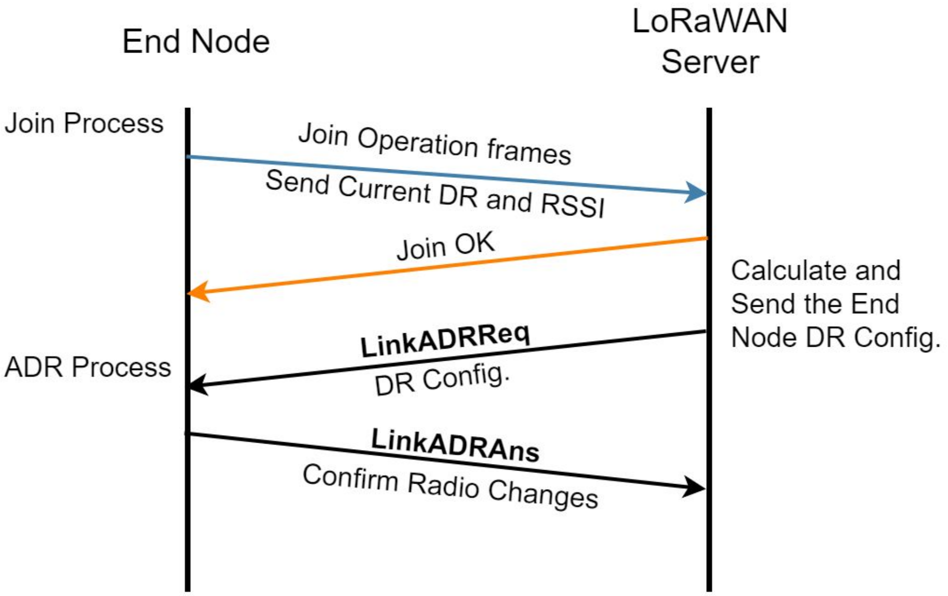
ttn-lw-stack-docker - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

```
  # Web UI configuration for "thethings.example.com":
  oauth:
    ui:
      canonical-url: "https://thethings.example.com/oauth"
      is:
        base-url: "https://thethings.example.com/api/v3"

# HTTP server configuration
http:
  cookie:
    block-key: "" # generate 32 bytes (openssl rand -hex 32)
    hash-key: "" # generate 64 bytes (openssl rand -hex 64)
  metrics:
    password: "metrics" # choose a password
  pprof:
    password: "pprof" # choose a password

# If using custom certificates:
# tls:
#   source: file
#   root-ca: /run/secrets/ca.pem
#   certificate: /run/secrets/cert.pem
#   key: /run/secrets/key.pem

# Let's encrypt for "thethings.example.com"
tls:
  source: "acme"
  acme:
    dir: "/var/lib/acme"
    email: "you@thethings.example.com"
    hosts: ["thethings.example.com"]
    default-host: "thethings.example.com"
```

Review on the LoRa ADR process

- The LoRaWAN Adaptive Data Rate process is a negotiation between the End Node and the Server of which Data Rate value the EN should use. The Data Rate is set by the Server, and it is changed in the Backoff sequence process.



Size (octets)	1	2	1
***LinkADRReq* payload**	DataRate_TXPower	ChMask	Redundancy

Table 17: *LinkADRReq* payload format

Bits	[7:4]	[3:0]
DataRate_TXPower	DataRate	TXPower

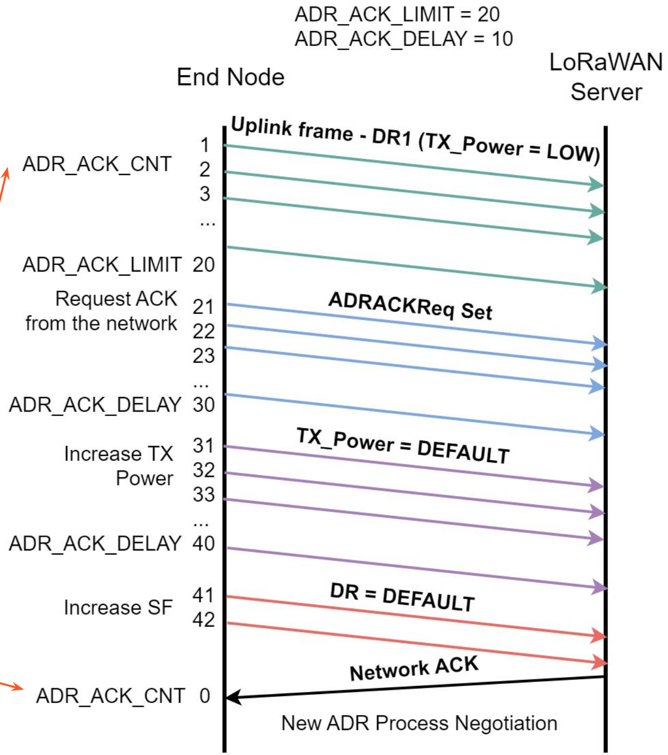
Table 18: DataRate_TXPower field format

Review on the LoRa ADR process

- If an end-device wishes to check for connectivity loss or if uses a data rate higher than its default data rate or a TX power lower than its default, the end-device SHALL periodically validate whether the Network is still receiving the uplink frames (from LoRaWAN link layer specs.).
- When the End Node stops receiving ACK frames from the Network, it must initiate a Backoff sequence, changing the power and Data Rate in steps.

ADRACKCnt	ADRACKReq	Data Rate	TX Power	NbTrans	Channel Mask
0 to 63	0	DR1	Max –9 dBm	3	Normal operations channel mask
64 to 95	1	No change	No change	No change	No change
96 to 127	1	No change	Default	No change	No change
128 to 159	1	DR0 (Default)	Default	No change	No change
≥ 160	1	DR0 (Default)	Default	1	For dynamic channel plans: re-enable default channels. For fixed channel plans: All channels enabled

Table 9: Example of a data rate backoff sequence

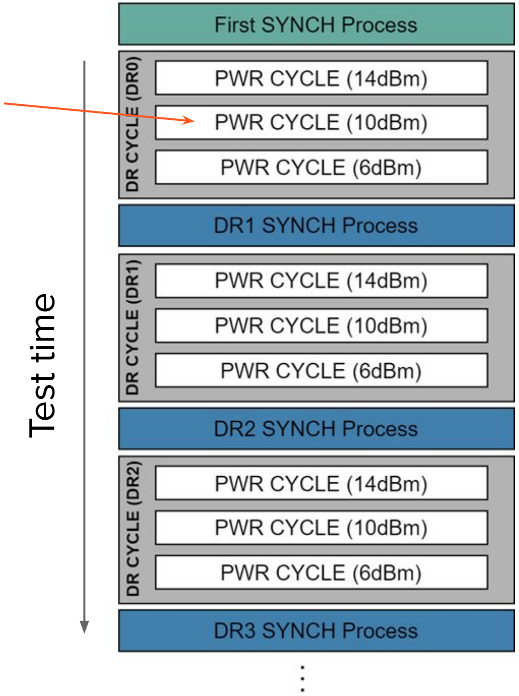


Considering a CLASS A device, the time to run the Backoff process may be extensive (If an uplink frame is sent every 30 mins, 40 frames would take 20h). ACK_LIMIT and ACK_DELAY must be properly set.

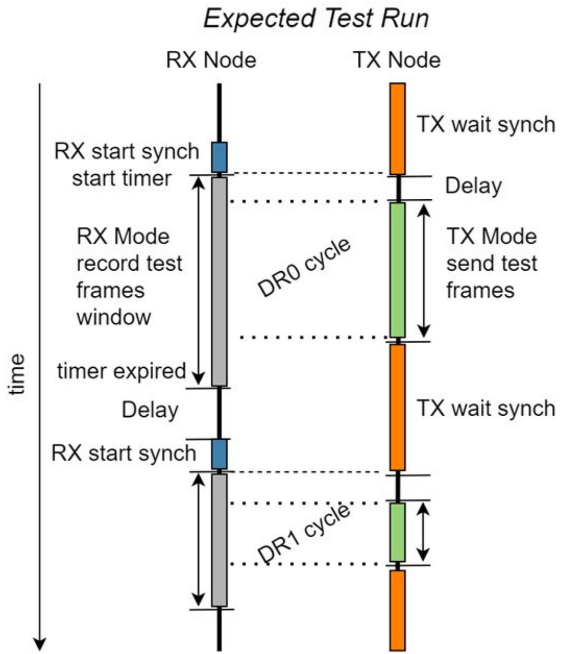
NUCLEO LoRa Testbench and extending the LoRaWAN Duty Cycle

- In the testbench develop for the NUCLEO WL55JC1, the LoRa Radio is configured in all the DR and TX_Power values in the Regional Parameters definition.
- To allow the test to run automatically, a SYNCHRONIZATION process was developed to allow the RX Node to be configured with the same Data Rate as the TX Node.

In each PWR CYCLE 200 frames are sent by the TX Node.

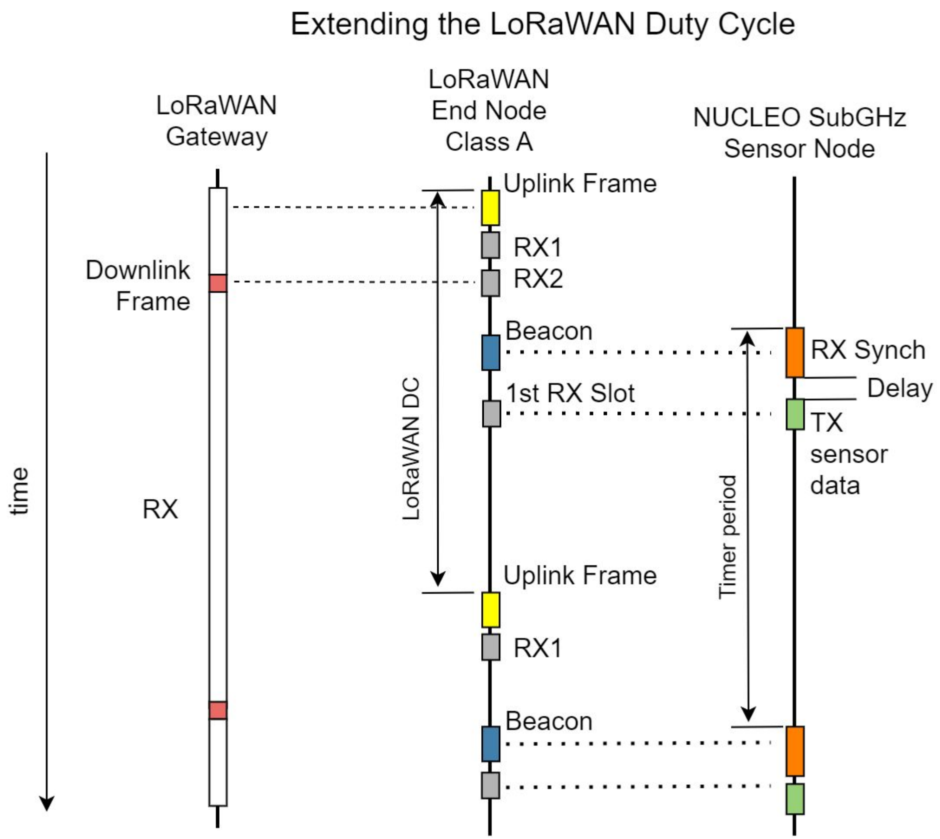


Testbench Synchronized Nodes



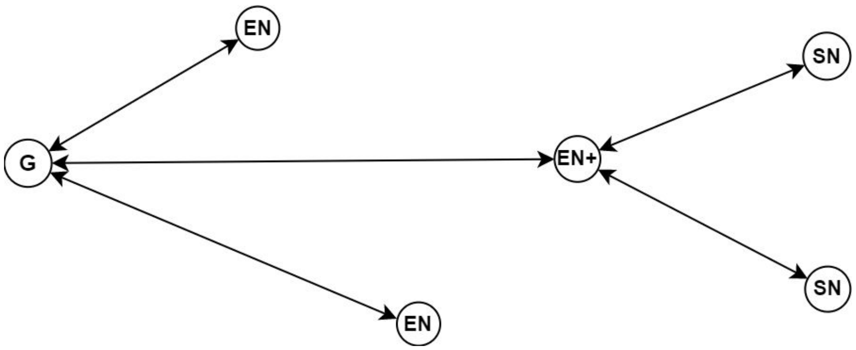
NUCLEO LoRa Testbench and extending the LoRaWAN Duty Cycle

- To extend the LoRaWAN Class A device Duty Cycle, the developed synch. process could be included in the End Node program. This would allow SubGHz Nodes (Nodes with simplified Radio configurations) to transmit frames to the End Nodes. The sensor information can be included in the LoRa frames sent to the network, extending the monitoring coverage.
- In this idea, the End Nodes would provide a periodic Beacon and divide the subsequent RX periods in time slots, using timers (similar to what was done in the Testbench).
- This process, however, would not support many extra SubGHz Nodes transmitting to a single End Node, unless a more complex MAC protocol is included (maybe a CSMA process).



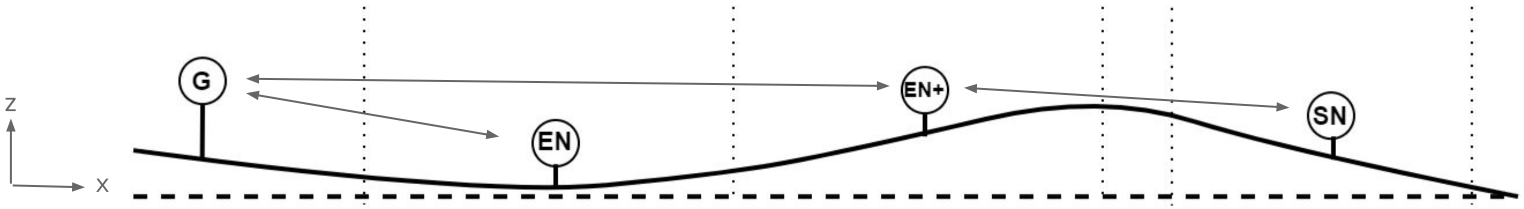
NUCLEO LoRa Testbench and extending the LoRaWAN Duty Cycle

- The WSN using this extended LoRaWAN DC would be organized like:

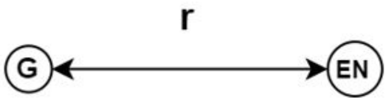


The SubGHz Nodes still use LoRa to transmit, but other Modulations could be explored as well

- As long as the “hidden” regions are small (and would not required more than a few SubGHz sensor nodes to be monitored), the LoRaWAN End Node extended should be able to receive the frames and include the data in its regular transmission.



A Model for evaluating the energy cost in the End Nodes when transmitting a data frame



Consider that an End Node (EN) has a function that describes the energy cost to transmit packets to a Gateway over a distance r as:

$$C(r)$$

Considering a WSN organized in a Star topology, and that the End Nodes transmit directly to the Gateway (Single Hop), The Cost function to transmit over a distance r is:

$$C_{SH}(r) = C_{TX}(N, p)$$

Where $C_{TX}(N, p)$ is the average energy cost to transmit a packet with a PDR of p and a maximum number of transmissions N defined by:

$$if\ p = 0, C_{TX}(N, p) = f(N)$$

$$if\ p = 1, C_{TX}(N, p) = f(1)$$

$$if\ 0 < p < 1, C_{TX}(N, p) = \sum_{n=1}^{N-1} [p(p-1)^{n-1} \times f(n)] + \left(1 - \sum_{n=1}^{N-1} [p(p-1)^{n-1}]\right) \times f(N)$$

A Model for evaluating the energy cost in the End Nodes when transmitting a data frame

$f(n)$ is the function that provides the amount of energy used to transmit n packets and is specific for the device that is being used (in general is a linear function). When using ADR, the bit rate and TX Power values must be added to compute the energy cost to transmit a frame, the function will be then: $f(n, TX_{Power}, DRx)$.

$$f(n, TX_{Power}, DRx) = n \times (TX_{Power} \times ToA)$$

$$in\ LoRa: ToA \propto DRx$$

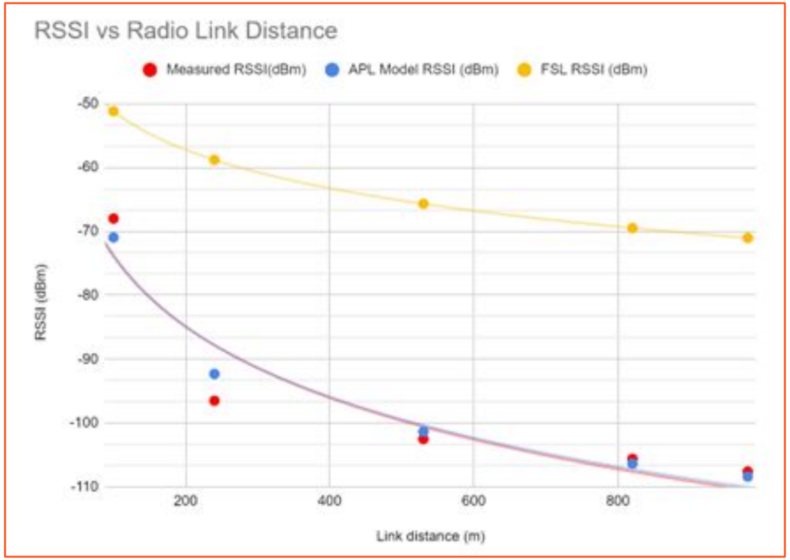
TABLE 1. ToA for Different Data Rates

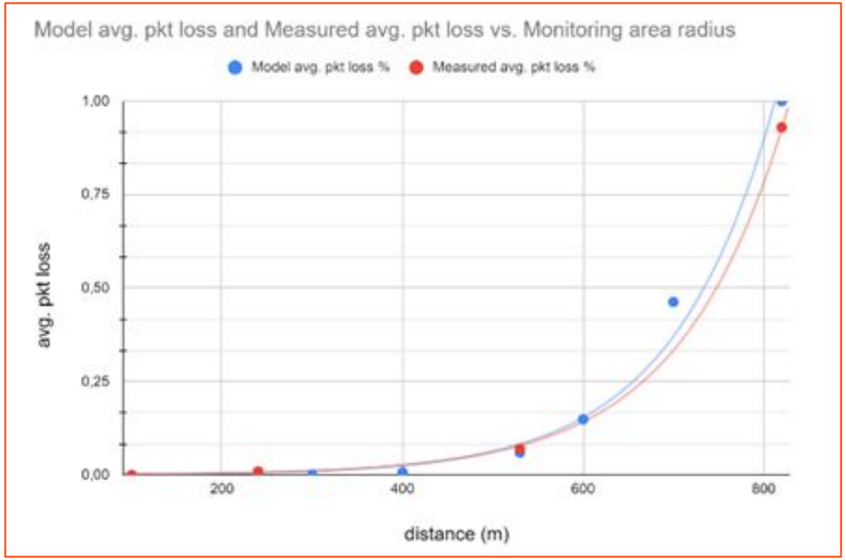
Data Rate	Configuration	Payload Size (bytes)	ToA (s)
0	LoRa:SF12/125 KHz	51	1.9087
1	LoRa:SF11/125 KHz	51	1.0363
2	LoRa:SF10/125 KHz	51	0.5591
3	LoRa:SF9/125 KHz	115	0.5868
4	LoRa:SF8/125 KHz	222	0.6006
5	LoRa:SF7/125 KHz	222	0.3412
6	LoRa:SF7/250 KHz	222	0.1706
7	FSK:50 Kbps	222	0.0371

A Model for evaluating the energy cost in the End Nodes when transmitting a data frame

The probability to deliver a packet p (PDR) in this WSN is the combination of two main factors:

1. the probability of a LoRa frame error caused by noise in the communication (a function of the SNR value in the Node): $P(SNR)$. The method to compute this error is very complex, an alternative is to use the Data collected from the **Testbench** to approximate an exponential function:





A Model for evaluating the energy cost in the End Nodes when transmitting a data frame

2. probability of collision of two or more frames sent by Nodes with the same channel and SF, as a function of the number of nodes transmitting to a single Gateway, considering a Poisson distribution of transmissions:

$$P(k) = 1 - e^{\frac{-2 \times ToA \times k}{DC_{TotalTime}} \times p_i}$$

$$p_i = \frac{1}{m_r \times n_r}$$

m_r *and* n_r are, respectively, the number of available channels and spreading factors in region r.

The PDR is then:

$$p = (1 - P(SNR)) \times (1 - P(k))$$

In a practical scenario, we can consider that the Duty Cycle time used in the WSN is much larger than the ToA of the individual frames, and that the number of nodes in the region is not too high. In this case the collision effect in the packet loss is minimal: $(1 - P(k)) \approx 1$.

A Model for evaluating the energy cost in the End Nodes when transmitting a data frame

By using the Testbench results, we can also estimate the Path Loss factor for the application field (open areas, rural environment):

$$A_{PL} = 10 \times \log\left(\Gamma \times \left(\frac{d_0}{d}\right)^n\right)$$

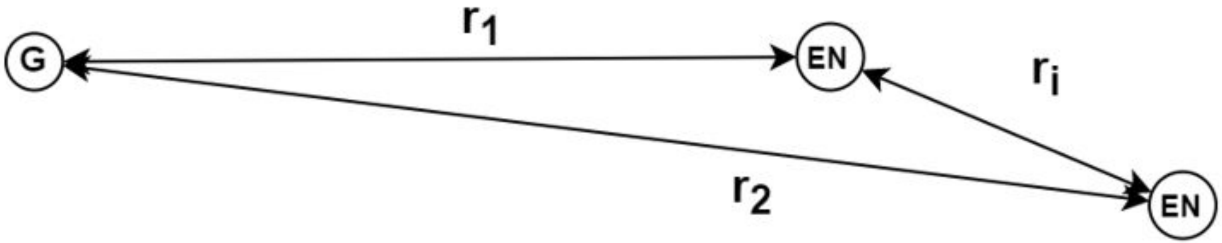
With the previous considerations we can compute the cost of sending a frame over a distance r, in a single hop: $C_{SH}(r)$.

Model resume:

1. With r we get A_{PL}.
2. With A_{PL} and the TX Power we get SNR.
3. With SNR we get p.
4. With p and $f(n)$ we get C (Energy cost to transmit over r).

A Model for evaluating the energy cost in the End Nodes when transmitting a data frame

With that we can move to a scenario where a far node could use an intermediate node to send its frames. In this case the frame would have to go through another hop to reach the Gateway:



Considering that $r_1 < r_2$, we can compare the Cost functions in a way that we evaluate if the use of multi-hop will improve the energy consumption of the system:

$$C(r_2) > C(r_1) + C(r_i)$$

A Model for evaluating the energy cost in the End Nodes when transmitting a data frame

By including the frame from r_i to the transmissions over r_1 we have:

$$C(r_2) = C_{SH}(r_2)$$

$$C(r_1) = C'_{SH}(r_1) + C_{RX}(k)$$

$$C'_{SH}(r_1) = C'_{TX}(N,p);\ f'(n)\ considering\ the\ extended\ frame\ size$$

Where $C_{RX}(k)$ is the cost function to receive a frame in an End Node that servers k SubGHz Nodes.

$$C(r_i) = C_{SH}(r_i)$$

Replacing:

$$C_{SH}(r_2) > C'_{SH}(r_1) + C_{SH}(r_i) + C_{RX}(k)$$

If the inequation is satisfied after replacing the calculated values, we can conclude that the multi-hop approach saved power to transmit frames in the system.

Outlook

- The deployment version of the LoRaWAN server must be configured. Once deployed, the Gateway will be connected to the server, and we can test sending messages from the NUCLEO board to the network.
- The Data should be available on the NodeJS front-end application that will display the sensor readings.
- On-field tests of the NUCLEO Testbench will provide the necessary data to complete the Cost model.