A minimalistic distance-vector routing protocol for LoRa mesh networks

Roger Pueyo Centelles[a,*], Felix Freitag[a], Roc Meseguer[a] and Leandro Navarro[a]

ARTICLE INFO

Keywords:
LoRa
Mesh
Multi-hop
Routing
LPWAN

ABSTRACT

LoRa has gained traction in the Internet of Things (IoT) domain as a Low Power, Wide Area Network (LPWAN) radio technology, providing low-power and long-range communication in the sub-GHz band. Most often it is used as part of the LoRaWAN architecture with a star-of-stars topology, but it can also be operated standalone, with a more flexible mesh network topology. In this paper, we propose a minimalistic Distance-vector (DV) Routing Protocol (RP) for LoRa mesh networks. It takes advantage of LoRa's multiple Spreading Factors (SFs), their trade-off between transmission distance and bit rate, and their orthogonality property. This way, it can use well-known metrics like Hop Count (HC), Expected Transmission Count (ETX), etc., and also more complex ones, like multi-SF-aware Time on Air (ToA). In the future, we aim at evaluating it both with the FLoRa framework and the OMNeT++ simulator, and with embedded boards featuring a LoRa radio transceiver.

1. Introduction

LoRa, which stands for *long range*, is a wireless communication technology owned by Semtech [1] that operates in the sub-gigahertz range of the radio spectrum. It employs Chirp Spread Spectrum (CSS), a proprietary modulation technique resistant to multi-path fading and suitable for noisy environments, aiming to provide low throughput communication with links of more than 10 km –outdoors, in rural areas– while maintaining low power consumption [1].

Several parameters of the LoRa physical layer can be configured in order to optimize communications for a given scenario or application, or even on a per-device basis: radio band and frequency, channel bandwidth, transmission power, Forward Error Correction (FEC) rate and SF. IoT deployments commonly operate on license-exempt Industrial, Scientific, Medical (ISM) bands, which change from one geographic area to another; diverse LoRa transceivers are available to operate in any of them. Inside these bands, in turn, many channels are available for up-link and down-link transmissions, with thinner or narrower channel bandwidths and different maximum transmission powers allowed. In addition to the robust CSS modulation, LoRa's FEC provides protection against interference on noisy links. Table 1 summarizes the aforementioned configurable parameters and lists their possible values.

The SF is perhaps the most interesting link configuration parameter in LoRa, as it determines a direct trade-off between *transmission speed* and *communication range*. Furthermore, two LoRa transmissions on the same channel using *different* SFs are quasi-orthogonal, meaning that both can be successfully demodulated by the respective receivers.

Table 1
Configurable parameters in LoRa transmissions

Configurable parameter	Values
Radio band	169, 433, 868, 915 MHz [a]
Bandwidth	62.5, 125, 250, 500 kHz [b]
Transmission power	14 dBm (EU), 27 dBm (USA)
Spreading Factor	6 to 12 [c]
FEC rate	4/5, 4/6, 4/7, 4/8

[a] Common frequency allocations for ISM in different regions worldwide; LoRa may also be used in licensed bands.
[b] Smaller bandwidths (7.8 to 41.7 kHz) are also supported, although rarely used.
[c] Certain bandwidth and SF combinations may result in too long transmissions for specific radio bands in which time-on-air limitations often apply.

This feature is leveraged by the gateway LoRa transceiver chips, which are extensively used in the LoRaWAN architecture, but can also be taken advantage of by regular end nodes.

A drawback for LoRa (and, in fact, for any radio technology operating in the sub- GHz part of the spectrum) is the legal duty cycle limitations imposed in ISM bands, which only allows a device to transmit on a given channel for a maximum of percentage of time (e.g., in Europe, duty cycle is 1 %). Sometimes, depending on the local regulation, this can be avoided if Channel Activity Detection (CAD) mechanisms are put into place.

2. State-of-the-art

Several proposals regarding multi-hop, mesh and routing for LoRa and LoRaWAN have made appearance in the recent years. They have been thoroughly classified and analyzed by different researchers from different points of view: taking the application scenarios into account [5], focusing on the LoRaWAN architecture [2], or on specific implementation aspects like topology and routing [4]. Their matu-

*Corresponding author
rpueyo@ac.upc.edu (R. Pueyo Centelles); felix@ac.upc.edu (F. Freitag); meseguer@ac.upc.edu (R. Meseguer); leandro@ac.upc.edu (L. Navarro)
ORCID(s): 0000-0002-3133-0657 (R. Pueyo Centelles); 0000-0001-5438-479X (F. Freitag); 0000-0002-9414-646X (R. Meseguer); 0000-0003-4775-5526 (L. Navarro)

[1] Semtech LoRa Technology Overview - https://www.semtech.com/lora

rity and Technology Readiness Levels (TRLs) are heterogeneous, and range from theoretical contributions to experimentally validated proposals in testbed or real-world deployments. Only few devices or solutions are commercially-available as end products. In this section we analyze the most relevant proposals, classifying them depending on their application domain, the scenario they tackle, or the specific features they provide.

2.1. Multi-SF reception with client node single-channel LoRa radio

The single-node multi-SF reception strategy was first exploited for building single-channel LoRaWAN gateways out of inexpensive ESP32-based end node devices [6]. While t

Kim et al. proposed an Adaptive Spreading Factor Selection (ASFS) scheme to build LoRa mesh networks using single-channel transceivers, increasing throughput and reducing costs [3]. Their proposal uses the modems' CAD capability with an iterative SF inspection and selection algorithm that allows links to operate independently at different data rates, achieving almost 100 % correct detection. This idea had been implemented already on single-channel gateways but had not previously been adapted for multi-hop usage.

Using ASFS allows nodes to choose different and faster SFs, achieving data rates four to six times faster than without it (when all the nodes stick to a common, network-wide slower SF).

3. A minimalistic routing protocol for LoRa mesh networks

In order to deploy and assess multi-hop LoRa networks, we implemented a proactive, hybrid Layer 2/3 (L2/3) DV routing protocol that takes advantage of this radio technology's specific characteristics. In particular, it benefits from LoRa's SFs different range and orthogonality properties, which allow for concurrent transmissions between different pairs of nodes, and introduces a novel multi-SF-aware ToA metric calculation that minimizes the total transmission time for a packet to reach the destination.

The protocol main features are:

- Distance-vector: best routes towards any destination are calculated in a distributed way across nodes, based on a cumulative metric.
- Concurrent, overlaid networks: thanks to the orthogonality properties of LoRa's SFs, several "virtual" layered networks can operate on the same radio channel, providing higher global throughput.
- ToA metric: route costs are calculated based on the end-to-end packet transmission time, taking multi-SF capability into account.
- Layer 2+3 hybrid: by aggregating the two layers into one, we simplify the architecture and the requirements for low-power embedded LoRa devices.

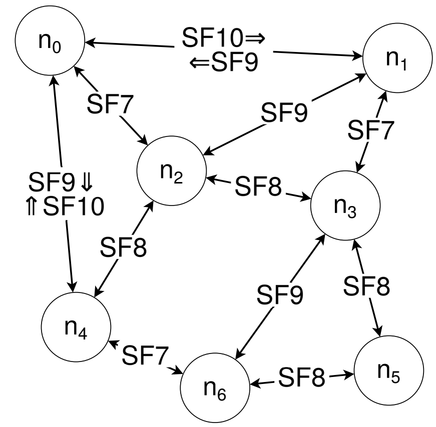

Figure 1: Topology of a sample LoRa mesh network. Links between each pair of nodes may use any valid SF (the diagram shows the smallest and fastest possible to create a working link). Most link are symmetrical, but others require different SFs on each direction; this may generate different routes in each direction between a pair of nodes.

- Pro-active: network nodes periodically broadcast available routes, independently from data traffic, refreshing routes and keeping them up-to-date and readily available.
- Duty cycle-aware: for networks operating in unlicensed ISM bands, time-on-air limitations in the form of duty cycles are usually enforced; these restrictions can be embedded into the route metrics calculations.
- Flexible and configurable: many aspects of the protocol (metrics, packets timing, etc.) can be fine tuned to fit specific use cases.

3.1. Network addressing and topology

The routing protocol builds a flat mesh network topology, with no hierarchical differentiation between nodes, regardless of their hardware characteristics or their role at the application level. To participate in the network, a node runs an instance of the routing protocol, exchanges routes with its neighboring nodes periodically and generates a local routing table.

Figure 1 shows the topology of a sample LoRa mesh network running the routing protocol. Since direct communication between a pair of nodes can occur at different SFs, they are indicated by the fastest (i.e., the lowest) SF possible. Communication between distant nodes that are not directly connected is made by multi-hop packet forwarding, using the routes calculated by the routing protocol. While most of the links in the diagram are symmetric, a few of them require different SFs in each direction to achieve a successful communication. This case can happen in scenarios scenarios with heterogeneous hardware or environmental conditions, either temporary or permanent. As a result, packets traveling between a given pair of nodes could use different routes in each direction.

To simplify the design of the routing protocol and the applications built upon, we merge the data link (Layer 2 (L2)) and the network (Layer 3 (L3)) layers into a single one (compared to, e.g., the WiFi+IP stack, where addresses of different types are used on each layer). This solution reduces overhead on network traffic and computing effort on the nodes, although it may limit direct interoperation with other networks (i.e., the Internet). In our design, network addresses take 2 bytes, ranging from 0x0000 to 0xFFFF and resulting in up to 65.536 usable addresses per network. We consider this specific addressing space size to be a compromise value between the ability to build a large mesh network, LoRa's throughput and range performance, and ease of implementation. However, it may be reduced or increased when implementing it on particular scenarios. For example, a smart metering network covering a city with hundreds of thousands of nodes may require a larger addressing space but, in this case, other strategies (e.g., network partitioning) could be more effective.

The routing protocol supports transmission of network packets in unicast, to a given single node using its unique network address, or broadcasting to all the neighbors at one hop using a broadcast address. For the latter, the highest address (0xFFFF) is reserved. Reception acknowledgments, packet retransmissions, etc. are left to be implemented in higher layers. The routing protocol does not handle them, to keep it as simple as possible and reduce its footprint on embedded nodes. Therefore, neither one-hop or end-to-end delivery are guaranteed at they routing level.

3.2. Time on Air (ToA) metric

The SF is a key element of the LoRa radio technology, as it poses a trade-off between the transmission reach and the time required to send a packet. Roughly, switching to a SF one step higher (e.g., SF7→SF8) doubles the transmission time (or halves the transmission speed, roughly), while increasing distance around $\times 1,4$. Throughput, or transmission time, also have a direct relation with the power required to transmit a packet, which is specially critical in battery-powered devices. This is usually the case in the context of LPWAN for the IoT domain, where radio channel occupation and power (i.e., energy) are scarce resources.

Taking the abovementioned restrictions into account, we propose a Time on Air (ToA) routing metric to evaluate the cost of a path towards a destination that aims at minimizing the total end-to-end time required to transmit a packet, from source to destination, and the radio channel occupation.

The ToA metric in our routing protocol calculates the path cost towards a node in function of the SFs used by the successive links between forwarding nodes, until reaching the destination. It ponders the cost of the hop $h_{l,m}$ between two neighbor nodes n_l and n_m as a power of 2:

$$\mathrm{h}_{l,m} = 2^{\mathrm{SF}_{l,m} - \mathrm{SF}_{min}} \tag{1}$$

where $\mathrm{SF}_{l,m}$ is the smallest SF required to successfully transmit between nodes n_l and n_m, and SF_{min} is the minimum SF available in the system. [2] Therefore, we calculate the cost of a path between two arbitrary nodes n_i and n_j, with H intermediate hops, as:

$$\mathrm{ToA}_{i,j} = \sum_{k=1}^{H} \mathrm{h}_k = \sum_{k=1}^{H} 2^{\mathrm{SF}_k - \mathrm{SF}_{min}} \tag{2}$$

where SF_k corresponds to the SF used in each of the intermediate H hops in the route. In case two or more paths are available with the same metric, the one with the next hop using a smaller SF is preferred. Still, in case of tie (same metric and same SF in the next hop), the path is chosen randomly among the contenders.

The example in Figure 1 uses SF7 as the smallest SF available (SF_{min}). Using the ToA metric, the cost of the direct single-hop path from node n_0 to node n_1 would be $2^{10-7} = 8$. Instead, the the three-hops path via nodes n_2 and n_3 would be preferred, since its ToA metric would be $2^{7-7} + 2^{8-7} + 2^{7-7} = 1 + 2 + 1 = 4$. The usage of a simple additive metric calculation is convenient in the context of resource-constraint IoT devices, which are often driven by 8 or 32 bit Microcontroller Units (MCUs). This way, these operations can be easily implemented and require few processor cycles.

3.3. Routing tables

Each node in the network runs an instance of the routing protocol, creating a local routing table that is constantly updated as messages from neighboring nodes are received. Being a DV protocol, the table consists of a list of all the nodes known to be in the network, the neighbor through which to reach them, the path cost and the route expiry time. With the information contained in the table, every node is theoretically able to communicate with any other node in the network, either directly (if they are neighbors) or indirectly using multi-hop. Furthermore, every node is also able to forward multi-hop traffic that goes through it towards its destination.

Besides the routes towards other nodes, each node also keeps track of the routes from neighbor nodes to it. Since the physical connection between two neighbor nodes may not be symmetrical, the routing protocol also needs to feed back neighbor nodes with the information of its inbound links, so that they can correctly calculate the routes towards it.

Table 2 shows a snapshot of the routing table for node n_0 from Figure 1. In the first block, the three entries on top show the direct, single-hop routes from node n_0 to its neighbors n_1, n_2 and n_4 (indicated by *Dest=NextHop*, but using a different SF). The next entries show multi-hop routes to other routes in the network (indicated by *Dest≠NextHop*). This includes the nodes that are not directly reachable using one hop (n_3, n_5 and n_6) and also those which have a better metric using a multi-hop path rather than the direct link (this is the case for n_1 and n_3). For instance, the direct single-hop path $n_0 \Rightarrow n_4$ has a higher ToA cost than the multi-hop path $n_0 \Rightarrow n_2 \Rightarrow n_4$.

[2] The smallest SF available in LoRa is SF6. For practical reasons, the minimum SF commonly used is SF7, but a higher value could be preferred.

Dest	NextHop	SF	Cost	Expiry
0x0001	0x0001	10	-	189
0x0002	0x0002	7	-	293
0x0004	0x0004	9	-	251
0x0001	0x0002	-	4	293
0x0003	0x0002	-	3	293
0x0004	0x0002	-	3	293
0x0005	0x0002	-	5	293
0x0006	0x0002	-	4	293
0x0002	0x0001	-	4	189
0x0003	0x0001	-	12	189
0x0004	0x0001	-	9	189
...	...	-	...	...
Dest	**Source**	**SF**	**Cost**	**Expiry**
0x0000	0x0001	9	-	189
0x0000	0x0002	7	-	293
0x0000	0x0004	10	-	251

Table 2
Routing table for node n_0 from Figure 1. The first block contains the routes from n_0 to all other nodes. The second block corresponds to routes from neighbor nodes to n_0 (required to properly account for asymmetric links).

The second block of entries in Table 2 shows inbound routes to node n_0. Node n_0 does not actually use them for its own routing decisions. However, it must keep track of them, and let its neighbor nodes learn about them. This way, neighbors can be aware of what is the minimum SF required to reach n_0 and correctly calculate their routes towards it.

Upon the successful transmission of a new routing packet, the receiving node processes it and refreshes its local routing table, updating the information accordingly. First, the link characteristics with the source neighbor node are added to the routing table, or refreshed if already present. Then, one by one, the announced routes are processed. Depending on the case, they will be added to the routing table, used to update or replace known routes, or discarded if not useful. The diagram in Figure 2 shows this process as a flow chart.

As exemplified by Table 3, a routing table may contain different entries with routes to the same node, via different next hops. While the routing protocol will use the one with the lowest cost by default, this provides alternatives in case the main route suddenly increases its cost or expires. However, to limit the growth of the table unnecessarily, a maximum number of total and per node routes can be established.

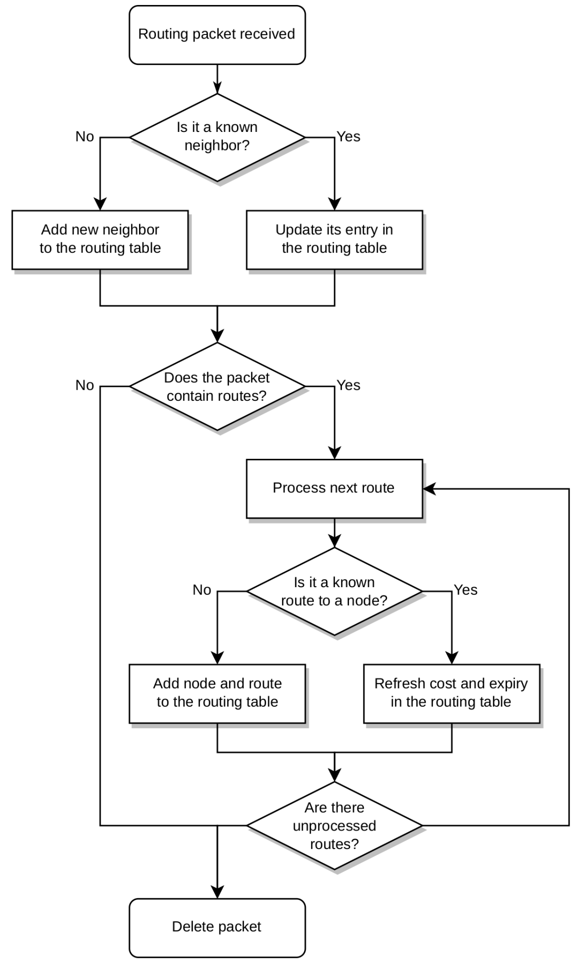

Figure 2: Flow chart upon reception of a new routing packet at a node.

3.4. Packets structure

Data and routing packets have a very similar structure, which is schematically depicted in Figure 3. Since packet size is limited by the LoRa hardware to a maximum of 256 B, their design uses a minimalistic approach to reduce overhead introduced by the header, which takes 7 B at most.

The first 4 B of a packet hold the address of the node originating the message (*Source*, 2 B) and the address of the final recipient (*Destination*, 2 B). The next field (*Via*, 2 B) is used in multi-hop data packets to indicate the address of the next hop in its route. Its value changes from hop to hop, as the packet is forwarded by intermediate nodes. In the last hop in the route, or for single-hop data packets, the *Via* field should be the same as the *Destination*. The *Flags, Time to live (TTL)* space (1 B) is reserved to tag packets if required by the application (2 b) and to track the TTL (6 b). Last, the *Payload* field holds the actual application data or the routing protocol exchanged between different nodes (see next section), holding up to 249 B.

Routing packets are differentiated from regular data packets because they use the broadcast address in the *Destination* field, instead of a unicast address. Therefore, the *Via* field is not necessary, and is hence omitted to save 2 B that are added to the payload, which becomes up to 251 B long.

3.5. Routes exchange between neighbor nodes

Nodes running an instance of the routing protocol periodically broadcast routing packets to their neighbors. This way, they proactively generate the network topology and keep it updated, by refreshing the routing tables locally as packets from neighbor nodes are received. These packets include:

- the source node information (i.e., its address)
- a 6 b incremental counter (instead of the TTL, which

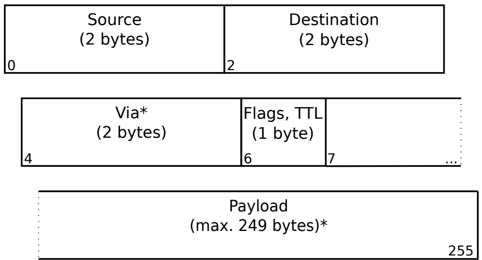

Figure 3: Structure of a data/routing packet. *Source* and *Destination* bytes identify the communication end points. The *Via* field is omitted when the broadcast address (0xFFFF) is set as the destination. The *Flags/TTL* byte is used to identify the packet type and watch its lifespan. Depending on the packet type, the *Payload* field can carry either the routing protocol information or the the application data.

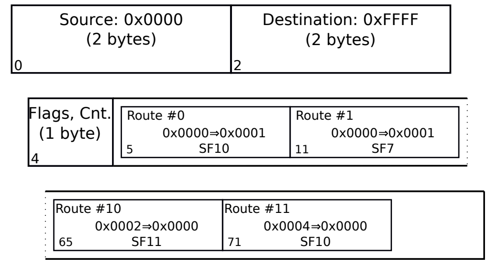

Figure 4: Sample routing packet as sent by node n_0 from Figure 1. The *Destination* address field indicates a broadcast packet destined to its neighbors. Routes in the payload correspond to those in the routing table from Figure 2.

can be used to evaluate packet loss and collisions on a link, to infer its quality or its occupation)

- an excerpt of the source node routing table (i.e., a list with the best routes and their path cost to the other nodes in the network).

With this information and the details from the LoRa radio physical layer (namely, the SF used to transmit the message) the nodes that receive the broadcast packet update their local routing tables accordingly. In turn, these nodes will propagate their routes further away, to their neighbors, when they send their own broadcast messages.

The devices running the routing protocol are expected to have single-channel LoRa radios, using the technique described in Section 2.1 to enable multi-SF reception. Nodes can send and receive packets on any SF available between SF_{min} and SF_{max}. However, transmissions on each SF have different reach and can be received by different nodes in the network. Therefore, routing packets need to be transmitted on all the SFs, to discover all the possible neighbors.

Because of the $2:1$ transmission time relation between two consecutive SFs, we design the routing protocol to broadcast messages on a given SF twice as often as on the immediately higher SF (i.e., broadcast packets using SF9 are sent at double the rate than packets using SF10). This compensates the cost of using different SFs, and also helps keeping routes using faster links more up to date than those using slower links. Broadcast packets on different SFs are sent in random order (rather than sequentially), using the following probability formula:

$$p(\mathrm{SF}_n) = \frac{2^{\mathrm{SF}_{max}-\mathrm{SF}_n}}{\sum_{i=min}^{max} 2^{\mathrm{SF}_i-\mathrm{SF}_{min}}}$$

$$\text{since:} \tag{3}$$

$$\sum_{i=min}^{max} p(\mathrm{SF}_i) = 1$$

$$p(\mathrm{SF}_{min}) = 2^{\mathrm{SF}_{max}-\mathrm{SF}_{min}} \cdot p(\mathrm{SF}_{max})$$

Sending packets on different SFs randomly, instead than using a predefined sequence, also helps avoiding repeated collisions if nodes become synchronized.

The following pseudo-code implements the random selection of a SF between SF_{min} and SF_{max} with the probability described above:

```
SF = 0;
while SF == 0 do
    thisSF = SFmin;
    while thisSF ≤ SFmax do
        if random(0, 1) < 0.5 then
            SF = thisSF;
            break;
        else
            thisSF++;
        end
    end
end
```

Algorithm 1: Random selection of the SF on which to broadcast a routing packet.

Routing packets are similar to regular data packets, except that they use the broadcast address as the destination and skip the *Via* field, as they are single-hop. Their payload contains the network routes, as advertised by the source node. Figure 4 show an example of a routing packet, (corresponding to node n_0 from Fig. 1), where the payload contains the best routes contained in the routing table from Table 2. In case the routing table does not fit inside the payload of a packet, routes to transmit are chosen randomly by assigning each of the routes a transmission probability, in function of its *Cost* value. This design decision makes routes to nearby nodes to be more updated than those to distant devices.

Nodes, upon reception of a new routing packet, process it and refresh their local routing table, updating it accordingly. First, the link characteristics with the source neighbor node are added or refreshed in the routing table. Then, one by one, the received routes are processed and added or updated in the routing table. The diagram in Figure 2 shows this process as a flow chart.

Scope	Parameter	Range	Default
LoRa	Minimum SF	SF7, ... SF$_{max}$	SF7
LoRa	Maximum SF	SF$_{min}$, ... SF12	SF12
Routing	Metric	ToA, HC, ETX, RSSI	ToA
Routing	Average routes broadcast period (SF$_{min}$)	$0 \ldots \infty$	60 s
Routing	Routes expiry time (SF$_{min}$)	$0 \ldots \infty$	300 s
Routing	Max. routes to a node	$0 \ldots 2$	2
Routing	Max. total routes	$0 \ldots 1024$	1024
Routing	Routing/Data traffic prior.	$n : 1$	$10 : 1$
Routing	Forward/Local traffic prior.	$n : 1$	$10 : 1$
Regulation	Duty cycle (%)	$0.1 - 100$	100

Table 3
Configuration options.

3.6. Protocol configuration

In order to adapt to specific use cases and conditions, many parameters of the RP can be configured. This section summarizes the most remarkable ones for the RP operation.

First, several LoRa-related aspects can be customized., like the maximum SF to use can be changed (from the default SF12 to a smaller one in order to ensure that one is never used, which could be required in certain regulatory domains). Instead of the ToA metric, other well-known ones can be chosen (HC, ETX, etc.). Other aspects, like the expiry time of a learned route can be modified, to ensure the liveliness of the information contained in the routing table. Table 3 shows these configurable parameters.

4. Future work

We aim at evaluating the presented minimalistic RP in two environments. First, in an emulated scenario, using the FLoRa framework and the OMNeT++ simulator. Second, in a realistic environment, using embedded microcontrollers featuring a LoRa transceiver.

References

[1] Augustin, A., Yi, J., Clausen, T., Townsley, W.M., 2016. A study of LoRa: Long range & low power networks for the Internet of Things. Sensors 16. URL: https://www.mdpi.com/1424-8220/16/9/1466, doi:10.3390/s16091466.

[2] Cotrim, J.R., Kleinschmidt, J., 2020. LoRaWAN mesh networks: A review and classification of multihop communication. Sensors 20. URL: https://www.mdpi.com/1424-8220/20/15/4273, doi:10.3390/s20154273.

[3] Kim, S., Lee, H., Jeon, S., 2020. An adaptive spreading factor selection scheme for a single channel LoRa modem. Sensors 20, 1008. doi:10.3390/s20041008.

[4] Osorio, A., Calle, M., Soto, J.D., Candelo-Becerra, J.E., 2020. Routing in LoRaWAN: Overview and challenges. IEEE Communications Magazine 58, 72–76. doi:10.1109/MCOM.001.2000053.

[5] Pueyo Centelles, R., Freitag, F., Meseguer, R., Navarro, L., 2021. Beyond the star of stars: An introduction to multihop and mesh for LoRa and LoRaWAN. IEEE Pervasive Computing , 1–10doi:10.1109/MPRV.2021.3063443.

[6] Westenberg, M., 2020. Single Channel LoRaWAN Gateway. URL: https://github.com/things4u/ESP-1ch-Gateway.