# On design and deployment of fuzzy-based metric for routing in low-power and lossy networks

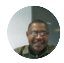
**Conference Paper** · October 2015

DOI: 10.1109/LCNW.2015.7365929



**3 authors**, including:


Patrick Olivier Kamgueu
University of Yaounde I



Thomas Djotio
École nationale supérieure polytechnique de Yaoundé

# On Design and Deployment of Fuzzy-Based Metric for Routing in Low-Power and Lossy Networks

Patrick-Olivier Kamgueu[†,§] Emmanuel Nataf[†], Thomas Ndie Djotio[§]
[†]University of Lorraine, UMR 7503, Nancy - France
INRIA Grand Est - MADYNES Team
firstname.surname@inria.fr
[§]University of Yaounde 1, Cameroon
LIRIMA, MASECNESS Team

***Abstract*—Minimizing the energy consumption and hence extending the network lifetime is a key requirement when designing an efficient sensor network protocol. QoS-aware routing in Wireless Sensor Network (WSN), aims to take into account other networks performance aspects as minimizing end-to-end delay (as well as jitter), reducing packet loss rate and minimizing the energy consumption of the network during data transmission. These objectives are sometimes conflicting therefore tradeoffs must be made between energy conservation and QoS considerations. The general problem can be reformulated as a Multi-Constrained Optimal Path problem (MCOP) which is known as NP-complete. The latter raises a real challenge as sensor nodes are very limited in resources capabilities. We propose to use fuzzy inference mechanism to seek a good tradeoff between all given metrics and constraints. This paper discusses the implementation of combining several routing metric using fuzzy logic to design a RPL objective function, the routing standard for the Internet of Things. The proposal is integrated on Contiki operating system, the deployment was performed on a real world indoor WSN. Obtained results show improvements compared to the common implementation of the RPL protocol and demonstrate relevance of our contribution.**

## I. Introduction

In recent years, Wireless Sensor Networks (WSN) as become an attracting field arousing interest of the scientific and industrial communities. Topology of such network is a collection of large number of sensor nodes deployed in the target area to detect some physical phenomenon. Usually a particular node called the *sink*, attached to a base station, receives all network data through neighbor nodes using multi-hop radio communication. Nodes are generally low-cost and low-power small devices that are equipped with only a limited data processing capability and low transmission rate, with a battery operated energy supply and scarce memory. The introduction of IP-based protocols and open standards led to the adoption of IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) and IPv6 Routing protocol for Low power and lossy networks namely RPL (pronounced ripple) [20], standardized by Internet Engineering Task Force (IETF). This opens new opportunities in various fields and application areas, such as home automation, smart cities, power grids, healthcare and critical area control (nuclear power plant, forest fire, disaster prevention, etc.).

Routing protocol is a key issues for WSN and RPL [20] was designed to take into account the unique characteristics of this kind of network. A number of metrics [19] are intended to be accounted by the protocol during the network topology building phase. These metrics are implemented as an objective function (`OF`) with the purpose of setting up various performance objectives for the network, in order to satisfy the requirements of the target application. So far, only two objective functions have been specified and standardized for RPL: The first uses hop-count as routing metric also called `of0` [18] and the other uses the expected number of transmission needed to successfully send a packet to its destination (ETX) namely `Minimum Rank with Hysteresis Objective Function` [6]. The use of other defined criteria (energy, latency, throughput, etc.) are left to the implementer. The possibility of combining several of those metrics into one, to ensure quality of service (QoS) or meet application requirements is not well-addressed.

Several traffic flows with different QoS requirements may share the same WSN, including for example both periodic and non-periodic data. For instance, one application can report periodically the temperature and lighting, while reacting to sudden events like threshold crossing or the entering of a person in a room. The set of services to be fulfilled when transmitting a stream of packets from source to destination are diverse, such as reduce end-to-end delay and maintain steady jitter, minimize packet loss, while extending the network lifetime. But these requirements are sometimes conflicting issues under a WSN context. For instance, one can improve reliability by increasing the number of maximum allowable retransmission or using higher transmission power levels. However, in both cases, more energy will be spent, thus reducing the overall network lifetime. So tradeoffs must be made between energy conservation and other QoS considerations. The problem becomes: how to make these tradeoffs at runtime or find integrated performance metrics that account the above requirements online. Traditional wired and wireless network QoS algorithms and techniques are not directly applicable on WSN due to their unique features and resource constraints.

Combining several routing metrics in RPL to improve QoS has already been investigated in the literature [11] and falls into two forms of combination: additive and lexicographic. In this paper we propose to take advantage of fuzzy logic to solve it and contribute to improving RPL by considering multiple objectives. This approach is promising because by using this paradigm, with a small memory footprint, we can seek for a halfway between several criteria, even antagonistic (which is not the case with the aforementioned combination methods).

A necessary condition with the additive composition is that basic metrics follow the same dimming direction (growing or decreasing). Lexicographic composition overcomes the latter restriction, but the main shortcoming with this approach is that in most cases when comparing two composite metrics (written as a vector), only the first metric component is considered and subsequent values are used to break the tie.

The remainder of the paper is organized as follows: Section 2 presents related works on combining metrics for routing in WSN, as well as works that use fuzzy inference system for routing design. Section 3 describes an overview of RPL standard. In section 4, we describe the proposed objective function design, followed by implementation parameters, experiment results and discussions in section 5. Finally, we conclude and discusses future directions in section 6.

## II. Related Work

In order to account for QoS in RPL, Karkazis *et al.* [11] propose to use additive and lexicographic composition to optimize more than one performance aspects. They also indicate conditions that basic metrics must hold to satisfy properties of convergence, optimality and loop freeness. In the additive approach, the composite metric $M$ is written as a linear combination of basic metrics ($M = \sum_i \alpha_i m_i$, where $m_i$ are basic metrics, each weighted by given coefficients $\alpha_i$). The main shortcoming of this scheme is that basic metrics must necessarily be defined on the same order relation, thus it restricts the type of metrics to be considered. In the lexicographic approach, given two basic metrics $M = \langle m_1, m_2, \ldots, m_k\rangle$ and $N = \langle n_1, n_2, \ldots, n_k\rangle$ with the respective order relation $\preceq_1, \preceq_2, \ldots, \preceq_k$, metric elements are evaluated sequentially: $M \preceq_{lex} N \Leftrightarrow (m_1 \preceq_1 n_1$ or $m_1 = n_1 \wedge m_2 \preceq_2 n_2, ...)$. So the subsequent metric values are accounted only to break the tie. Hence most of the time many metrics are not considered.

In that context, Link Estimation and Parent Selection protocol [22] combines hop-count and Link Quality Indicator (LQI) in lexicographic manner to select the best next hop. Firstly, the source node selects neighbouring with the minimum hop towards the sink node as its parent. If there are more nodes having the same hop-count value, it's the one with the largest LQI that is chosen. The main disadvantage of this protocol is the early death of nodes and the unbalanced energy dissipation. RPLRE [21] overcomes the latter and suggests to use a probability selection scheme that take into account the residual energy when choosing next hop, besides LQI and hop-count. The probability scheme improves selection process and avoids choosing the same node most often. The result is a more balanced energy consumption among potential parents and the network lifetime is thus extended.

Aslam *et al.* [2] propose a composite metric that uses multiple parameters or QoS constraints to find an optimal route for the Optimized Link State Routing (OLSR) protocol. OLSR protocol is natively based on hop-count and the composite value is computed as a linear combination of maximum available bandwidth, minimum delay and jitter. Contrary to previous methods, EARQ [9] a novel routing protocol for wireless industrial sensor networks, considers basic metrics criteria separately, instead of a unique combined metric value. This scheme aims to provide real-time, reliable delivery of a packet while considering energy awareness. Firstly, the path with a lower energy cost is selected according to a probabilistic model. In addition, only paths that may deliver packets in time are selected to achieve real-time requirement. Moreover, source node may send a redundant packet via alternate path if the reliability is not met. EARQ supposes that every node knows its location and rely on a GPS mechanism or a location process for that. However this assumption is not always feasible for many WSN deployments.

There is a growing interest for the integration of artificial intelligence techniques or control processes like fuzzy rule-based systems to design protocols for WSN. GAFO [5] uses a genetic adaptive fuzzy hop selection scheme to make optimal choices for robust packet transmission in WSN involved in varying channel conditions. Authors describe a fuzzy system engine that takes signal to noise ratio and outage probability as input, to determine the possibility of a neighbor node to be selected as the next hop for data forwarding. Experiments show that under the same conditions, this protocol outperforms the crisp approach on average by 20% for reliability and 15% for total energy consumption. Likewise, a cluster head election method using fuzzy logic has been introduced by Gupta *et al.* [7] to overcome the defect of LEACH [8], a popular cluster head selection technique. The main idea for the LEACH protocol is that nodes are elected depending on a stochastic model and use localized clustering. The consequence is that some cluster heads may be very close to each other or may be located at the edge of the WSN. This inaccurate cluster heads distribution could not maximize energy efficiency. Other fuzzy-based schemes [3], [17] were proposed to improve election process involved on LEACH. These have proven that the network lifetime can be efficiently extended using fuzzy variables (concentration, energy and node centrality). Unfortunately, LEACH is not applicable to networks deployed in large regions, since it uses single-hop routing where nodes can transmit data directly to the cluster head, afterwards the latter transmits data to the sink.

Unlike LEACH where each cluster head must directly send data to the sink, EDARP [23] establishes a *Fuzzy Spanning Tree* that uses the energy and distance to construct a routing path over all cluster heads. These two criteria are used to generate a *fuzzy election number* and lead to the selection of the best parent into the routing tree. Hence, energy consumption is balanced among all nodes by keeping rotation in cluster head election and parent node selection. FEAR [1] proceeds slightly differently. Rather than using a clustering mechanism to build the hierarchical topology, FEAR protocol directly builds a logical tree topology among all network nodes. A ranking-based system that relies on fuzzy inference is used, so that nodes rank their neighbors by considering both neighbor depth and power consumption. This fuzzy ranking system is used to construct and maintain the tree topology. Compared to RPL, FEAR generates more control messages which implies a greater power consumption. In addition, the protocol uses a node identification (ID) construction model where a node's ID is computed based on the node's parent ID. Unfortunately, when a parent node dies, all nodes in its subtree must recompute their ID as soon as a new parent is found, involving more processing and communication overhead.

## III. RPL OVERVIEW

RPL [20] is a distance vector routing protocol optimized for low power and lossy network, where multipoint-to-point is the dominant traffic pattern. The protocol also supports point-to-multipoint traffic pattern using destination advertisement mechanism, and provides a basic structure for point-to-point route. The network topology is organized as one or more Destination Oriented Direct Acyclic Graph (DODAG), each rooted at a single point that acts as sink for the topology. Three new types of ICMPv6 messages are defined and manipulated:

- DODAG Information Object (DIO) used to create and maintain upward routes.
- DODAG Destination Advertisement Object (DAO) used to install downward routes.
- DODAG Information Solicitation (DIS) actively used by a node wishing to join the network or asking for up to date information.

The topology building starts at the root (initially the only router which is part of a DODAG), that sends DIO messages in its neighbourhood. This message conveys all common configuration parameters, including root ID, mode of operation, timer values etc. Upon reception of a number of such messages, neighbour nodes may participate in the DODAG according to the Objective Function (`OF`), select theirs parents and then start to issue their own DIO messages. This process spreads gradually to cover the whole network while new nodes join the DODAG. Only one node among the parent set (*the preferred parent*) acts as the next-hop on the path towards the root.

RPL pro-actively creates and maintains the topology by regularly sending ICMP control messages in the vicinity. The spreading rate is governed by the trickle algorithm [12], that reduce the overhead induced by control messages. This is done by sending DIO less often when the topology is steady, but reacts and quickly spreads information on topology change or when inconsistencies are detected.

An important point is when a given node receives more than one *consistent* DIO, each from a different neighbour and must choose which one would be the preferred parent. This choice is governed by the objective function that specifies how this node selects its best parent into the parent set, and calculates its own rank (a gradient representing its relative position with respect to the root) from the parent's rank. Different criteria also called routing metrics [19] are defined to capture link or node characteristics on the path for parent selection purpose. The rank computation is derived from the set of these selected metrics, and must monotonically decrease as we move towards the root. This last property enables the routing structure to maintain its acyclic nature, thus helps to avoid routing loops.

Unlike existing `OF`s [6], [18] that rely on a unique routing metric to construct the DODAG, we want to combine several metrics into RPL by taking into account more than one performance aspect. The IETF ROLL working group has left open to the implementer the definition of new `OF`s, only the acyclic nature of the graph must be preserved. Routing metrics and constraints have been proposed by the working group, but the possibility to combine them is also not well-addressed. The next section shows how we design such `OF` using fuzzy inference system.

## IV. ROUTING METRIC DESIGN

### A. Fuzzy Inference System

Fuzzy logic reasoning allows us to transform several input variables (delay, ETX and energy) into one (Quality). The fuzzy inference system consists of several steps:

- Fuzzification: take a crisp value input and determine its degree of membership (fuzziness) for the appropriate fuzzy sets.
- Fuzzy inference: Apply combination rules to *"fuzzified"* inputs and compute a fuzzy output.
- Aggregation: If an output depends on more than one rule, this step unifies all values into one.
- Defuzzification: Convert the fuzzy output obtained at the previous step into a crisp value.

In this paper, due to its simplicity and efficiency, we use the most common fuzzy inference method, namely Mamdani model [13].

### B. Composite metric design

To illustrate fuzzy inference reasoning, we consider the network topology depicted in figure 1. Node `N`, in order to send data to the final destination `S`, must select the next hop (either `P1` or `P2`) as the preferred parent. This choice is governed by received information as shown. We would like to know which parent is the most suitable next hop based on those criteria (energy, delay and ETX), according to the fuzzy inference engine.

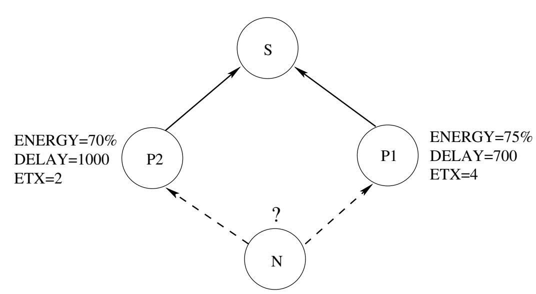



Fig. 1: Parent Selection Process

*1) Linguistic variables:* Node's performance knowledge is represented as linguistic variable. In this case, we consider the following three metrics (or linguistic variables in the fuzzy inference vocabulary):

- ETX - The expected number of required transmissions before a packet reaches the destination. It assesses the transmission accuracy.
- Delay - The average time for a packet to reach its destination.

- Energy - The energy cost of the path, also energy of the node having the smallest remaining battery level on the path.

We use cross-layer mechanisms to retrieve ETX and delay from data link and network layers. To estimate ETX, a sending node records the number of transmission before receiving an acknowledgement. The average value is computed as an EWMA (Exponentially Weighted Moving Average) over the time. The one-hop delay is computed as the required time to send a packet and receive the acknowledgement. This time includes MAC contention mechanisms. The overall end-to-end delay is the delay up to the sink.
Node's energy is estimated based on the online energy estimation model [16] and its implementation we conducted in [10], [14]. The model takes into account the energy spent under a constant current load at each node state (transmitting, receiving, idle and sleeping). We also consider the `recovery effect` which assesses the energy recovered during inactivity period.

*2) Fuzzification process:* To avoid the complexity of directly combining the selected three linguistic variables into one, we perform the fuzzification process in two stages, as shown in figure 2.

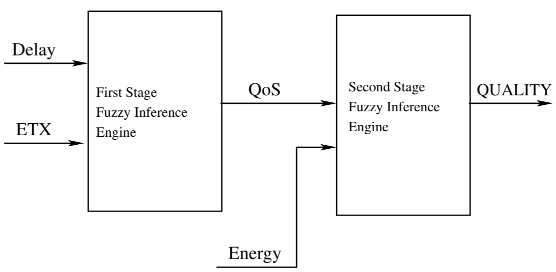



Fig. 2: Fuzzy Inference Engine

*First stage of fuzzification:* On the first stage, we combine delay and ETX as inputs to compute QoS, which is taken in its turn as input for the next stage. The linguistic variables used to represent delay are divided into `short, average` and `long` membership functions and ETX fall into `small, average` and `high`. Figure 3 depicts their membership functions normalized by the number of hops ($hc$) upwards to the sink, since delay (respectively ETX) is computed at the path level. Table I illustrates the relationship between these two linguistic variables for the computation of QoS. This table is built according to the desired rules, easy to express, and which depend on the knowledge of an expert. For instance, a long delay and high ETX implies a very slow QoS. Therefore shorter is the ETX and smaller is the delay, better is the QoS to consider.

| ETX / Delay | short | average | long |
|---|---|---|---|
| **small** | very_fast | fast | moderate |
| **average** | fast | moderate | slow |
| **high** | moderate | slow | very_slow |

TABLE I: QoS Output Metric

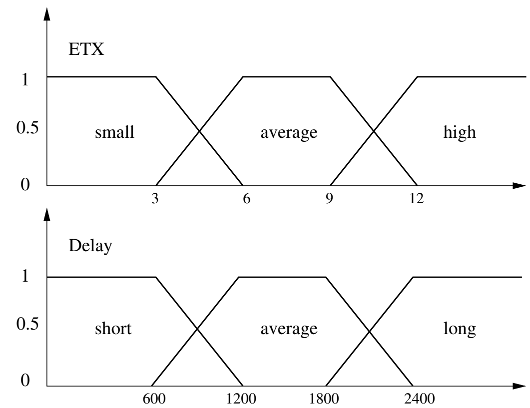



Fig. 3: One-hop membership functions

For instance, considering a crisp value ETX, formula 1 indicates how the level of membership is in the `average` fuzzy set, at one hop ($ie.\ \ hc = 1$). Similar formulas establish the level of membership for other ETX fuzzy sets (`small` and `high`), as well as delay and energy linguistic variables.

$$average(etx) = \begin{cases} 0 & \text{if} \quad etx \leq 3 \\ \frac{etx-3}{6-3} & \text{if} \quad 3 < etx < 6 \\ 1 & \text{if} \quad 6 \leq etx \leq 9 \\ \frac{etx-12}{9-12} & \text{if} \quad 9 < etx < 12 \\ 0 & \text{if} \quad etx \geq 12 \end{cases} \tag{1}$$

For the example provided in figure 1, node `N` computes as level of membership `small(etx)=0.66, average(etx)=0.33` and `high(etx)=0` for the parent node `P1`. The same types of computations for `P1`'s delay allow us to determine the respective fuzzy sets as `short, average` and `long`, with values `0.83, 0.16` and `0`.

Since the QoS relates to ETX and delay, the previously computed membership functions are combined according to table I. The Mamdani model allows us to use the $minimum$ operator as the composition function, and $maximum$ for the aggregation operator. For instance, formula 2 indicates how to compute `moderate(QoS)` fuzzy set from inputs. In this case, three rules are fired, we aggregated them with the $maximum$ operator. Likewise, we establish formulas for $QoS$ fuzzy sets ranging from `very_fast` to `very_slow`.

$$moderate(QoS) = \max \begin{pmatrix} \min(high(etx), short(delay) \\ \min(avg(etx), avg(delay)) \\ \min(small(etx, long(delay)) \end{pmatrix} \tag{2}$$

For our illustrative topology in figure 1, node `N` computes three non-zero QoS membership functions concerning neighbour `P1: very_fast(QoS)=0.66, fast(QoS)=0.33`, and `moderate(QoS)=0.16`. These values are defuzzified (as described in §IV-B3) into a single QoS output (`QoS=0.78`), and then used in the next fuzzification stage.

*Second stage of fuzzification:* As the second stage of the fuzzy inference system, we combine the previously computed QoS with the energy linguistic variable to provide `QUALITY`. For a given node, energy could be `low, medium` or `full`, and the output values for `QUALITY` is divided into seven levels ranging from `awful` to `excellent`. Table II shows how to derive `QUALITY` based on QoS and energy.

| QoS / Energy | low | medium | full |
|---|---|---|---|
| **very_slow** | awful | bad | average |
| **slow** | bad | degraded | average |
| **moderate** | degraded | average | acceptable |
| **fast** | average | acceptable | good |
| **very_fast** | average | good | excellent |

TABLE II: QUALITY Output Metric

*3) Defuzzification process:* All fuzzy values obtained after the aggregation step are converted into a single crisp output. The most common and accurate defuzzication method uses the centroid, where the result is the center of gravity of the polygon drawn using fuzzy values of the output membership function. Figure 4 illustrates the defuzzification process for `QUALITY` linguistic variable and formula 3 shows how to compute the final crisp value `Q`.

$$Q = \frac{\sum_{i=1}^{k} \alpha_i \times QUALITY(\alpha_i)}{\sum_{i=1}^{k} QUALITY(\alpha_i)}, \tag{3}$$

where $k$ is the number of fired rules, $\alpha_i$ is the domain value related to the $i^{th}$ rule, and $QUALITY(\alpha_i)$ is the level of trustiness value according to this corresponding domain. Output values range from 0 to 100 and indicate what is the level of quality to choose a neighbour as the next hop, according to the selected metrics. For the proposed topology, three membership functions:
`acceptable(QUALITY)=0.25`,
`good(QUALITY)=0.70`,
`excellent(QUALITY)=0.30`,
are fired as `P1`'s `QUALITY` output. The center of gravity for the depicted region is `77`. Similar computations produce value `70` for `P2`. Therefore the best next hop for `N` according to the fuzzy inference engine on these three criteria (ETX, delay and energy) is `P1`. By considering the lexicographic composition of metrics ETX, delay and energy respectively, this would have led to choose `P2` as the next hop.

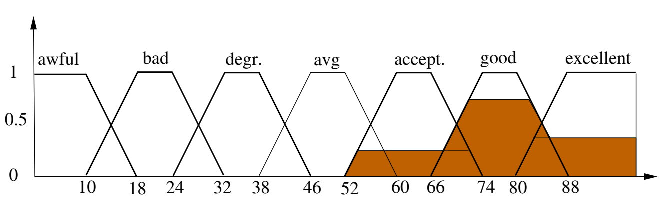



Fig. 4: QUALITY defuzzification

## V. EXPERIMENTS RESULTS

### A. Network setup

To evaluate the proposed combined metric, we deployed a real indoor WSN of twenty-eight sensor nodes as depicted by figure 5. They are placed in fourteen offices, two nodes per office, the displayed labels are node's ID and links indicate the next hop chosen by the node at a given time. Nodes are TelosB MTM-CM5000-MSP type, equipped with MSP430 16-bit Texas Instruments micro-controller, CC2420 radio frequency chips, various sensors (temperature, relative humidity and light sensor), one USB interface and operates on two AA Batteries. RF power range between 0 through -25 dBm (software configurable) and allows emission range from 20 to 30m in indoor environment. Sensor nodes, located in fixed place without mobility, sense data and transmit them to the sink for processing every 5 minutes. Sink node is directly attached to the gateway and thus is assumed to have unconstrained battery power. Nodes run Contiki version 2.7, a well-known operating system for embedded devices [4]. Table III summarizes other parameters of the protocol stack.

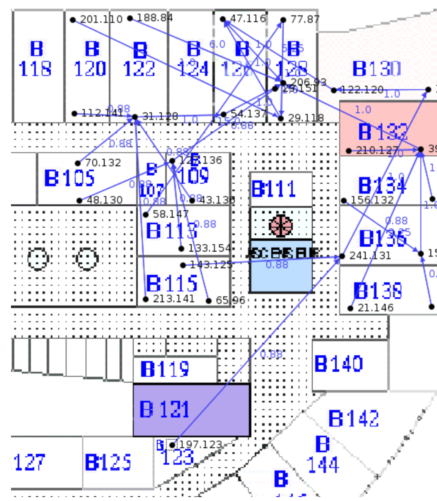



Fig. 5: Node placement

| Settings | Values |
|---|---|
| Application Layer | Periodic Sensor Data collection |
| Transport Layer | UDP |
| Network Layer | $\mu$IPv6 + 6LoWPAN + RPL (routing) |
| MAC Layer | non-persistent CSMA |
| Radio Duty Cycle | ContikiMAC |
| PHY + Radio chip | IEEE 802.15.4 w/ CC2420 |

TABLE III: Protocol stack

### B. Performance evaluation

We developed two scenarios of a data collection application, where each node sense its vicinity and transmit measured environmental parameter data (temperature, brightness and humidity) to the sink through neighbors using multihop links, during slightly more than two weeks. In the first scenario, the network is organized according to the standard RPL that uses ETX as the single routing metric. The second scenario instead uses the proposed combined metric scheme: ETX, Delay and Energy according to the Fuzzy Inference Engine. We are interested to evaluate the reliability of the application to collect and send data toward the sink, evaluate the stability of the routing by assessing the number of best parent changes over the time and finally the network lifetime by looking at energy depletion of nodes.

*1) Packet Loss Ratio:* As depicted by figure 6, experiment results indicate that the combined metric scenario obtain better results in packet loss ratio than the native ETX RPL. While the former stabilizes the loss rate around $5\%$, in the latter this loss is almost three times higher. When looking at the network startup phase, we can notice that the proposed scenario behaves better. Indeed, the combined metric based RPL remains relatively stable, whilst during the first three days, there is a high loss ratio for the ETX based routing (about 25%). This behavior of the ETX-based routing during the first three days is due to the bad routes selected and shows its slowness to reach to a steady state.

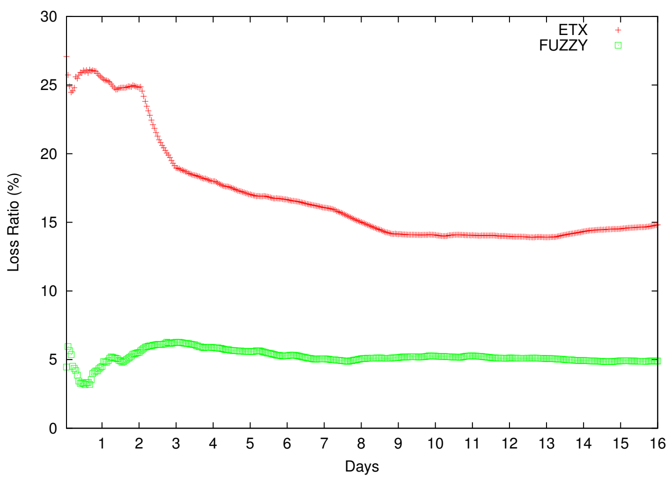



Fig. 6: Packet loss ratio over time at sink node

*2) Routing Stability:* This performance metric allows us to assess the number of best parent changes in the network over time. A high variation rate reveals an unstable topology and is not desirable, since it can impact the packet delivery and routing table flapping. As shown in figure 7, the number of best parent changes remains relatively low and rather steady over time for the fuzzy-based scenario, compared to the ETX-based scenario. The picture also displays the linear regression of number of parent change distribution. We can see that the number of parent changes per hour is higher, with an average of $43.52$ for the native ETX-based metric, whereas the same average is $6.63$ for the combined metric experiment.

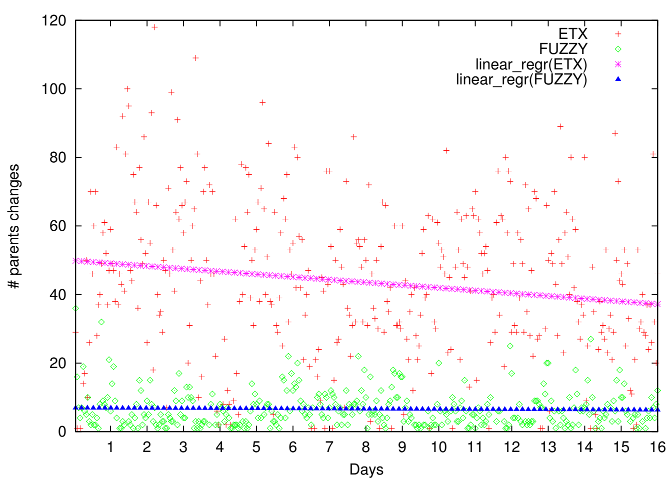



Fig. 7: Number of parent changes per hour

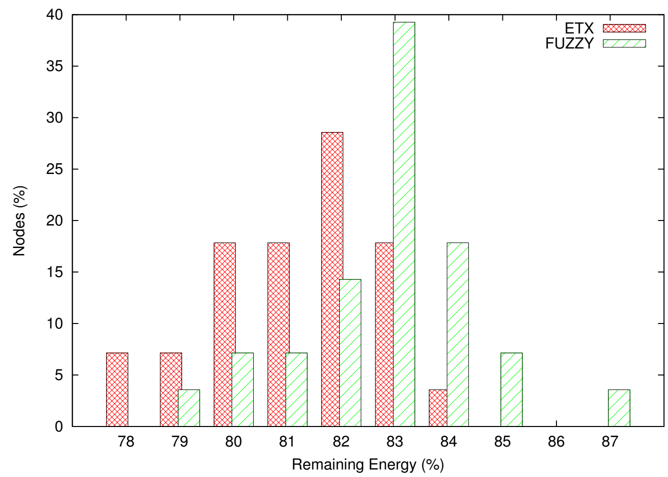



Fig. 8: Remaining power distribution

*3) Average Remaining Power:* All nodes start with the same power level (100% battery power level). Figure 8 depicts the remaining power distribution of nodes after sixteen days of experiments. It is clear that the pyramid induced by the drawn histogram is more to the left side for the fuzzy-based metric than the ETX scenario. Thus, the proposed combined metric is more energy conservative than the native ETX metric. More specifically, in the combined metric scenario, 67.85% of nodes have their energy higher than 83% of the initial battery level, while this proportion is only 21.42% for the ETX-based routing. Moreover, the combined metric scenario maintains a great number of nodes (39.27%) at a same high power level (83%). Likewise, we can denote that weak power nodes (having their remaining battery power level $\leq 81\%$ of the initial power level) in the ETX based scenario (50% nodes) outnumber the combined metric scenario (17.85% nodes). This shows that the proposed combined metric is more power efficient than the native ETX based routing, since it accounts battery level in addition to other criteria.

*4) End-to-end delay:* Delay is another metric that we consider when selecting the best next hop according the three selected parameters (in addition to energy and ETX). The real challenge to evaluate end-to-end delay in the real WSN deployment lies in the fact that all nodes must be synchronized or share the same reference clock. This requires the implementation of additional mechanisms that would be very expensive in terms of memory and processing resources, which already are very limited on the deployed sensor nodes. For these reasons and for simplicity purposes, we emulate the deployed WSN on similar conditions using Cooja [15], a popular WSN simulator environment. Nodes share the same image code as the live deployment, however they also share the same clock with Cooja, which enables us to accurately evaluated the end-to-end delay. Figure 9 shows the cumulative distribution function (CDF) of end-to-end delay. Although the gap between the two developed scenarios is not very large, the proposed fuzzy-combined metric behaves better. For instance, we can see that in the fuzzy-based metric, 75% of packets have a delivery time less or equal to 1s, where this proportion is 68% for the ETX-based metric. Since fuzzy-based scenario have a better delivery ratio as shown in §V-B1, the computed end-to-end delay in this scheme is more accurate than ETX-based ones, because more packets are accounted.

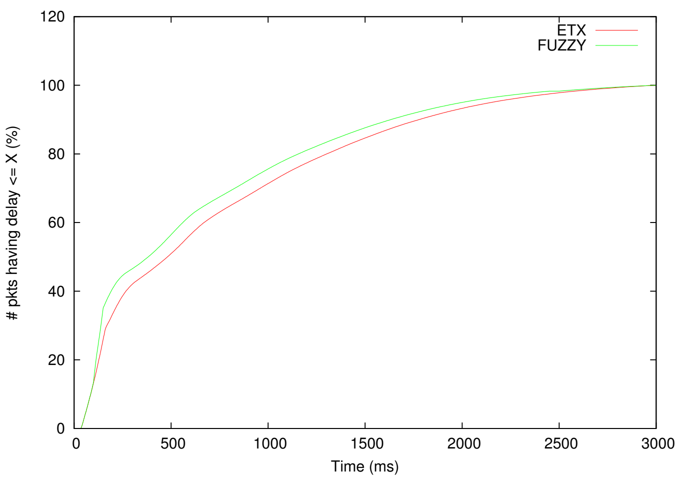



Fig. 9: CDF of end-to-end delay

## VI. Conclusion

This paper describes the design and implementation of a new RPL objective function that combines several metrics to optimize more than one network performance aspects. The proposed solution uses fuzzy inference system to merge expected transmission count, delay and node's remaining power into one unique value. We performed a real sensor network deployment in indoor environment to assess the metric. Experiments and results show that the proposed combined metric outperformed the ETX based routing on packet loss ratio, routing stability, energy efficiency and even on end-to-end delay. The proposed fuzzy inference system is used to find a *tradeoff* between input metrics which can be antagonistic. In our future work we aim to find a way to favor some metrics compared to others, thus better tune the contribution of each metric. Moreover a network can generate several traffic flows which don't have the same QoS requirements. For instance, some flows must comply with strict end-to-end delay requirements, while others must meet the transmission accuracy. It would be very interesting to investigate how to account those flows with regard to their requirement and take those into account at runtime.

## References


[1] I. M. Almomani and M. K. Saadeh, "Fear: fuzzy-based energy aware routing protocol for wireless sensor networks," *Int. Journal of Communications, Network and System Sciences*, vol. 4, pp. 403–415, 2011.

[2] N. Aslam, W. Phillips, and W. Robertson, "Composite metric for quality of service routing in olsr," in *Proc. of Canadian Conference on Electrical and Computer Engineering*, Niagara Falls Ontario, Canada, May 2004, pp. 759 – 762.

[3] M. Basirnezhad and M. N. Torshiz, "Energy efficient cluster head election using fuzzy logic in wireless sensor networks," *Int. Journal of Computer Science and Information Security*, vol. 9, no. 5, pp. 255–260, May 2011.

[4] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors." in *Proc. of LCN*, 2004.

[5] D. S. Ghataoura, Y. Yang, and G. Matich, "Gafo: genetic adaptive fuzzy hop selection scheme for wireless sensor network," in *Proc. of 5th IEEE IWCMC*, Leipzig, Germany, June 2009.

[6] O. Gnawali and P. Levis, "The Minimum Rank with Hysteresis Objective Function," RFC 6719, Internet Engineering Task Force, Sep 2012.

[7] I. Gupta, D. Riordan, and S. Sampalli, "Cluster-head election using fuzzy logic for wireless sensor networks," in *Proc. of 3rd IEEE Annual Communication Networks and Services Research Conference*, Halifax, Canada, May 2005.

[8] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. of 33rd Hawaii International Conference on System Sciences (HICSS)*, 2000, pp. 3005–3014.

[9] J. Heo, J. Hong, and Y. Cho, "Earq: Energy aware routing for real-time and reliable communication in wireless industrial sensor networks," *IEEE Trans. on Industrial Informatics*, vol. 5, no. 1, pp. 3 – 11, Feb. 2009.

[10] P. Kamgueu, E. Nataf, T. Djotio, and O. Festor, "Energy-based metric for the routing protocol in low-power and lossy network," in *Proc. of 2nd Sensornets*, Barcelona, Spain, Feb. 2013.

[11] P. Karkazis, P. Trakadas, H. C. Leligou, L. Sarakis, I. Papaefstathiou, and T. Zahariadis, "Evaluating routing metric composition approaches for qos differentiation in low power and lossy networks," *Trans. of Wireless Networks*, vol. 19, no. 6, pp. 1269–1284, August 2013.

[12] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle Algorithm," RFC 6206, Internet Engineering Task Force, Mar 2011.

[13] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Transaction on Computing*, vol. C-26, no. 12, pp. 1182 – 1191, 1977.

[14] E. Nataf and O. Festor, "Online estimation of battery lifetime for wireless sensors network," in *Proc. of 2nd Sensornets*, Barcelona, Spain, Feb. 2013.

[15] F. Osterlind and A. Dunkels, "Cross-level sensor network simulation with cooja," in *Proc. of 31st IEEE Conf. SenseApp*, Tampa, Florida, 2006, pp. 641 – 648.

[16] J. Rahmé and N. Fourthy, "Energy management for battery-powered embedded systems," in *Proc. of IEEE WCNC*, Sydney, Apr. 2010, pp. 277 – 324.

[17] G. Ran, H. Zhang, and S. Gong, "Improving on leach protocol of wireless sensor networks using fuzzy logic," *Journal of Information & Computational Science*, vol. 7, no. 3, pp. 767–775, 2010.

[18] P. Thubert, "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)," RFC 6552, Internet Engineering Task Force, Mar 2012.

[19] J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks," RFC 6551, Mar 2012.

[20] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, Internet Engineering Task Force, Mar 2012.

[21] C. Yan, J. Hu, L. Shen, and T. Song, "Rplre: A routing protocol based on lqi and residual energy for wireless sensor networks," in *Proc. of International Conference on Information Science and Engineering (ICISE)*, Nanjing, China, Dec. 2009, pp. 2714 – 2717.

[22] T. Yan and L. Sun, "Principle and performance evaluation of routing protocol in tinyos," *Trans. on Computer Engineering*, vol. 33, no. 1, pp. 112 – 114, Jan. 2007.

[23] M. Zeynali, L. M. Khanli, and A. Mollanejad, "Edarp : novel energy and distance-aware routing protocol in wireless sensor network," in *2nd ICIS : Information Technology, Culture and Human*, IEEE, Ed., Seou, Korea, Nov 2009.