

Using Machine Learning to Estimate the Optimal Transmission Range for RPL Networks

Moussa Aboubakar
CEA, LIST, Communicating
Systems Laboratory
F-91191 Gif-sur-Yvette
moussa.aboubakar@cea.fr

Mounir Kellil
CEA, LIST, Communicating
Systems Laboratory
F-91191 Gif-sur-Yvette
mounir.kellil@cea.fr

Abdelmadjid Bouabdallah
HEUDIASYC UMR 7253,
CS 60319
Compiègne Cedex 60203, France
bouabdal@utc.fr

Pierre Roux
CEA, LIST, Communicating
Systems Laboratory
F-91191 Gif-sur-Yvette
pierre.roux@cea.fr

Abstract—Nowadays, IoT low power networks have been largely adopted in various scenarios such as the factory of the future, smart agriculture, smart cities, and so forth. These networks allow the interconnection of resource-constrained devices (e.g. actuators and sensors) using wireless communication links. In such networks, having energy-efficient routing protocols is necessary in order to maximize the network lifetime. However, many factors may influence the energy consumption in IoT low power networks. These factors include the transmission power of devices, the data routing strategy, etc. In this paper, we address the effect of the transmission power on energy consumption in RPL networks (RPL : IPv6 Routing Protocol for Low-Power and Lossy Networks) and propose a new solution based on a multi-layer perceptron (MLP) model, to estimate the optimal transmission range of dynamic IoT low power networks deployed following a 3D topology. The results of our experiments show that our MLP model performs well compared to other machine learning models.

Index Terms—IoT, RPL, Machine Learning, transmission range.

I. INTRODUCTION

IoT low power devices may be deployed in a wide range of IoT scenarios such as smart cities, smart factories and smart agriculture. These devices are limited in terms of resources (e.g. memory, CPU, battery etc.) and communicate over error-prone and lossy radio channels. In this context, it is important to have efficient data routing schemes to maximize the network lifetime [1].

One of the commonly used protocols for data transmission in IoT low power networks is the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [2]. This routing protocol has been defined and standardized by the IETF ROLL Working Group and it allows performing data transmission among low power devices in wireless networks using different routing strategies, each defined by an objective function. In the literature, various improvements of RPL objective functions have been proposed in order to improve the network performance [3]–[5]. Nonetheless, those proposals do not take into account the transmission range of devices. In fact, a non-optimal transmission range may significantly degrade the network performance. To solve this issue, various approaches such as [6], [7] proposed a per hop optimal transmission range evaluation. Such distributed approaches may result in networks

with disjoint segments since they do not provide a global view of the network topology. Therefore, we propose a solution based on a supervised learning technique in order to estimate the optimal transmission range of RPL networks. This work extends our previous work [8] where we proposed a solution to determine the optimal transmission range for an RPL network deployed in a 2D environment with a fixed number of nodes. In particular, our current solution is based on the multilayer perceptron (MLP) and it enables the estimation of the optimal transmission range of dynamic RPL networks (where a node can be added or removed) following a 3D topology (e.g. a smart building). This solution is built using a set of data composed of various network topologies and their associated optimal transmission range.

II. OPTIMAL TRANSMISSION RANGE EVALUATION IN IOT LOW POWER NETWORKS

The selection of an optimal value of the transmission range in wireless sensor networks is challenging since short or long transmission range values cannot guarantee a good network performance [9]. In RPL networks, as shown in [8] and [10], the transmission range impacts the frequency of the update of routing tables. In fact, having a short transmission range may generate excessive energy consumption by intermediate hops in a multi-hop communication network. Likewise, having a long transmission range may require an important energy consumption for data routing. Hence, it is important to define an optimal transmission range in an IoT low power network in order to ensure a good network performance.

In [11], the authors proposed an energy efficiency metric for periodic monitoring applications called the Energy Distance Ratio per bit (EDRb). The proposed metric enables to determine the transmission range that minimizes the energy consumption in wireless sensor networks (WSNs). Similarly, Deng et al. [7] proposed a per hop method to determine the transmission range that minimizes the network energy consumption in a WSN. However, since there is no global view on the network topology, the transmission range determined by these methods may result in a network with isolated subsets of devices.

In [12], the authors show that determining the optimal transmission range of sensor nodes is NP Hard. Therefore,

in this proposal, we use a deep neural network algorithm to build a model that determine with a high accuracy, the optimal transmission range of IoT low power devices. In particular we used RPL networks deployed following a 3D topology. Our proposed model is designed to operate on a central entity (e.g. RPL root) which is aware of the network topology.

III. DESCRIPTION OF OUR PROPOSAL

One of the biggest challenges in IoT low power networks is to ensure a good performance of the routing protocol. For this purpose, it is important to firstly determine the optimal transmission range of network devices communicating using a wireless medium.

In a real deployment environment, IoT low power devices are generally deployed following a 3D topology and their number may vary over the time due to device failure, addition or removal of devices to/from the network and so forth. These events lead to an update of the network topology. However, in order to achieve a good network performance, the optimal transmission range for the network topology should be defined [7]. Considering that, in the current solution, we use the MLP algorithm to devise a model that can accurately estimate the optimal transmission range of dynamic RPL networks deployed in 3D environments. MLP is a class of neural networks which allows to perform regression or classification tasks by using a set of historical data [13]. In this work, we use MLP to perform a regression task. More specifically, we use the MLP algorithm to determine the relationship between a given network topology and its optimal transmission range. In fact, MLP algorithm is adapted to model complex relationships between a set of inputs and a set of outputs as shown in [14].

In order to obtain the set of data to build our MLP model, we proceeded to extensive simulations of various network topologies in an $X \times Y \times Z$ area on Cooja simulation tool [15]. For the sake of experimentation, we chose $X = 100m$, $Y = 50m$ and $Z = 200m$. The difference among the simulated topologies lies in nodes positions (coordinates) and the network size. For a given network topology, we varied the transmission range values in order to obtain the one that minimizes the network energy consumption while maintaining the network connectivity. We repeat this process for various network topologies sizes. In our simulation, we varied the number of devices from 30 to 100 with a step of 5. For each simulated RPL network, we incrementally varied the transmission range from 40 to 100 with a step of 10. In total, we performed 10500 simulations on an OpenStack virtual machine comprising 24 CPUs running at 2.299 GHz each. The RAM size of the virtual machine is 16 GB and each simulation lasts 1800s. At the end of the simulation, we obtained a dataset composed of 1500 entries. Each entry is composed of the optimal transmission range value and the associated RPL network topology. An example of the configuration of a given node deployed following a 3D topology in Cooja is given in Figure 1.

In this solution, the inputs of the MLP is the network topology (or coordinates of network devices) and the output is

```
<note>
<breakpoints />
<interface_config>
  org.contikios.cooja.interfaces.Position
  <x>7</x>
  <y>23</y>
  <z>63</z>
</interface_config>
<interface_config>
  org.contikios.cooja.mspmote.interfaces.MspClock
  <deviation>1.0</deviation>
</interface_config>
<interface_config>
  org.contikios.cooja.mspmote.interfaces.MspMoteID
  <id>1</id>
</interface_config>
<motetype_identifier>z12</motetype_identifier>
</note>
```

Fig. 1: Example of configuration of Cooja node deployed in 3D environment.

the optimal transmission range of the associated RPL network. However, since the input of a given MLP should have a fixed size whereas our dataset contains network topologies with various sizes, we used a padding technique in order to generate our MLP model. Clearly, we fixed L_{max} as the length of the input of our MLP so that when the input vector length is shorter than L_{max} , the shorter input is padded to L_{max} . Therefore, zeros are added to the input vector in order to get an input vector of size L_{max} .

Since MLP has many parameter values that need to be set before training in order to exhibit a good performance, we use the Tree-structured Parzen Estimators (TPE), a sequential model-based optimization (SMBO) [16] to obtain those values because it is not straightforward to perform that task manually due to the fact that the good hyperparameters have to be found in a large set of possible values. SMBO performs several trials of hyperparameters and incrementally uses better hyperparameters by applying Bayesian reasoning and updating a probability model (the probability representation of the objective function) [17].

Concretely, TPE performs the search of the parameters that give the best performance for a given machine learning model by modeling probabilities $p(x|y)$ and $p(y)$, which we can use to obtain $p(y|x)$ via Bayes rule. x represents hyperparameters and y the associated quality score. Therefore, $p(x|y)$ represents the probability of the hyperparameters given the score on the objective function and it is defined as follows [17]:

$$p(x|y) = \begin{cases} l(x) & \text{if } y < y^* \\ g(x) & \text{if } y \geq y^* \end{cases} \quad (1)$$

where $l(x)$ represents the distribution of parameter configurations yielding high score, while $g(x)$ represents the distribution of the other configuration. y^* is the quantile γ of the observed y values.

$p(y|x)$ is the surrogate probability model expressing the probability of y (actual value of the objective function using hyperparameters x) given x (hyperparameters) [17]:

$$p(y|x) = \frac{p(x|y) \times p(y)}{p(x)} \quad (2)$$

In this work, we assumed that the network is managed by a central entity (called the network manager), a device

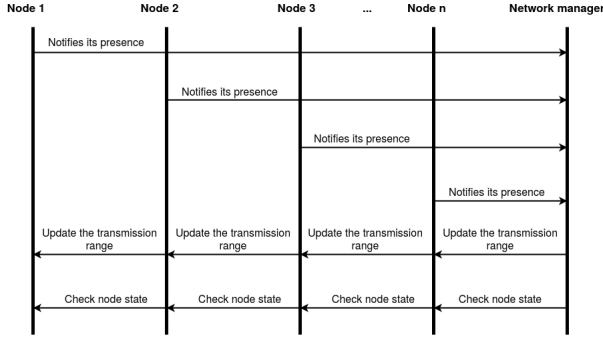


Fig. 2: Topology discovery.

with a powerful capacity, which could be the RPL root. In order to reconfigure the transmission range of network devices, the network manager should be aware of the network topology. The process of discovering the network topology is depicted in Fig. 2. Each device in the network notifies its presence to the network manager via a message. Then, the network manager will determine the optimal transmission range using our machine learning model and finally initiate the process to update the network transmission range. Algorithm 1 shows an overview of the process executed at the network manager. The training of our model is performed by the network manager which is also responsible for updating the transmission range of network devices. As shown in algorithm 2, each network device is responsible for sending its position to the network manager so that it can receive and update of its transmission range. We assumed that network devices are resource-constrained (in terms of memory, CPU and battery).

Algorithm 1 Process at the network manager

```

1: build MLP model using collected data
2: wait for a network topology (*)
3: if (new topology is received) then
4:   use MLP model to estimate the optimal transmission range
5:   send the optimal transmission range to network devices
6:   go back to (*)
7: end if
  
```

Algorithm 2 Process at the network device

```

1: while energy of device is not exhausted do
2:   send its position to the network manager (*)
3:   wait for the optimal transmission range
4:   if transmission range is received then
5:     update the transmission range
6:   end if
7:   if device position is updated then
8:     go back to (*)
9:   else
10:    wait device position update
11:    go back to (*)
12:   end if
13: end while
  
```

IV. PERFORMANCE EVALUATION AND DISCUSSION

A. Results

To evaluate the performance of our MLP model, we considered the Mean Absolute Error (MAE) and the Mean Square Error metrics [18]. MAE refers to a metric corresponding to the

TABLE I: MLP parameters

Parameters	Value
Type of activation function	relu, linear
Loss function	MAE
Batch size	90
Number of epochs	800
Dropout regularization	0.5
Number of the hidden layer	3
Number of neuron in hidden layer	300
Training optimization algorithm	Adam

expected value of the absolute error loss and MSE is a metric corresponding to the expected value of the squared (quadratic) error or loss. Our learning and validation experiment is performed using Keras [19] and scikit-learn [20] libraries that allow to design machine learning models. We used the values of hyperparameters obtained using TPE algorithm to build our MLP model. The values of those hyperparameters are depicted in Table I. We evaluated the performance of our MLP model in three steps:

Step 1: We splitted our dataset into k -fold and used each fold once for validation and the remaining folds for the training. In our experimentation, we set the value of k to 6. We compared the performance of our MLP model with some state of the art machine learning models based on the same training and validation set. Figure 3 and 4 shows the spread of the MAE and MSE accross each cross validation fold for each algorithm. These results show that our MLP exhibits a good performance compared to other machine learning models such as linear regression, support vector regression and regression tree. This proves that our MLP model can estimate well the optimal transmission range compared to other machine learning models.

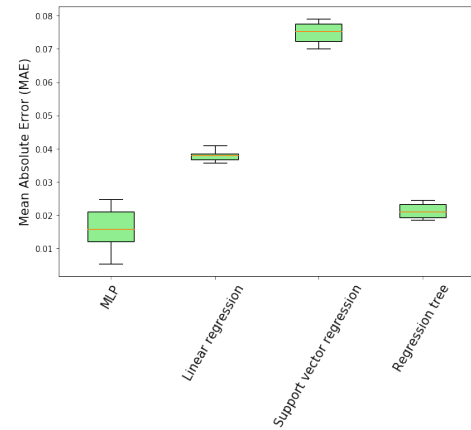


Fig. 3: Comparison of the Mean Absolute Error of different machine models.

Step 2: We randomly splitted our dataset obtained through simulation in two subsets as follows: 80 % for training and 20 % for the test. After that, we reduced successively 10% and 20% of the output of our training set in order to obtain a dataset with missing values. By doing so, we can observe the performance of our MLP model when using a dataset

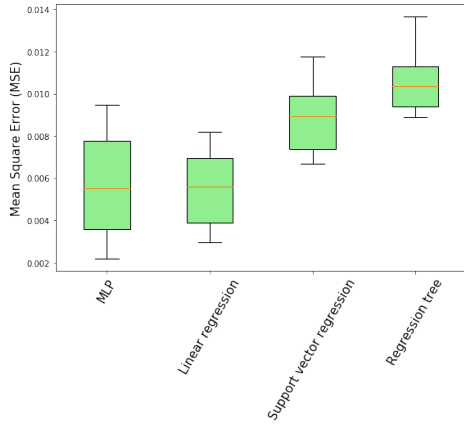


Fig. 4: Comparison of the Mean Square Error of different machine models.

with a reduced size during the training. In fact, for a given machine learning model, the size of the training set may impact the performance of the model. Additionally, in real world it is not straightforward to create 10k network topologies and identify their optimal transmission range as we did during the simulations.

We employed two strategies to deal with the remaining training set. The first strategy consisted in predicting the missing output values using the linear regression. In the second strategy, we simply removed the entries containing the missing output. We obtained different types of training dataset namely: D1 and D4, the training set composed respectively of 10% and 20% of predicted values. D2 and D5, the training set composed respectively of 10% and 20% of deleted rows. D3 is the training set without missing values. We trained different MLP models using those training dataset and the hyperparameters in Table I. Table II shows that models based on the training set with missing rows provide MAE values close to the MAE of the model based on training set without missing values. This means that our model can provide a good performance in a real world with a reduced training dataset.

TABLE II: Comparison of the MAE of MLP for different training sets.

Training set	MAE
D1 (dataset with 10 % of predicted values)	0.10
D4 (dataset with 20 % of predicted values)	0.10
D3 (dataset without missing values)	0.02
D2 (dataset with 10 % of deleted rows)	0.01
D5 (dataset with 20 % of deleted rows)	0.02

Step 3: Concerning the energy efficiency of our model, we performed various simulations of a number of network topologies on Cooja using the transmission range estimated by the different machine learning models. Each simulation lasted 1800 s and we evaluated the network energy consumption generated when using the transmission range estimated using the different machine learning models. Figure 5 shows that it is not always possible to obtain a minimal energy consumption

using our MLP model. This limitation is due to the error (difference between the predicted and the real value) obtained during the training of our model.

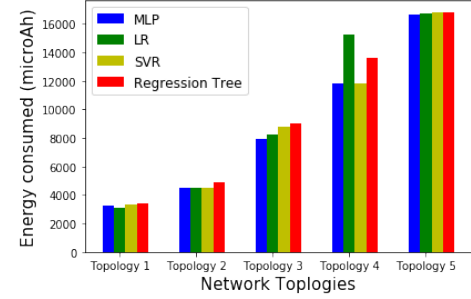


Fig. 5: Comparison of the network energy consumption generated by the transmission range of different machine learning models.

B. Discussion

The training of our model is performed by the network manager which is also responsible for updating the transmission range of network devices. We adopted this strategy in order to deal with the resource constraints of devices in RPL networks.

Solving the transmission range problem using a centralized approach requires attention on how the solution will react to dynamic networks (e.g. because of node mobility or node failure). In addition, the task of updating the transmission range cannot be neglected or considered as straightforward since it may present several problems on a real network with lossy channels.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented an MLP model that allows to determine the optimal transmission range in RPL networks where devices are deployed following a 3D topology. The results of our experiment show that our model performs well compared to other existing machine learning models such as linear regression, support vector regression and regression tree. In our future work, we plan to explore more machine learning models to design a solution to estimate a per cluster of nodes optimal transmission range. Our aim is to improve the energy consumption of our solution.

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their valuable feedbacks that helped us in better improving the content of the paper.

REFERENCES

- [1] A. A. A. Ari, A. Gueroui, B. O. Yenke, and N. Labraoui, "Energy efficient clustering algorithm for wireless sensor networks using the abc metaheuristic," in *2016 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 2016, pp. 1–6.
- [2] T. Winter, "Rpl: Ipv6 routing protocol for low-power and lossy networks," 2012.

- [3] Z. Wang, L. Zhang, Z. Zheng, and J. Wang, "An optimized rpl protocol for wireless sensor networks," in *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2016, pp. 294–299.
- [4] O. Iova, F. Theoleyre, and T. Noel, "Using multiparent routing in rpl to increase the stability and the lifetime of the network," *Ad Hoc Networks*, vol. 29, pp. 45–62, 2015.
- [5] X. Yang, J. Guo, P. Orlik, K. Parsons, and K. Ishibashi, "Stability metric based routing protocol for low-power and lossy networks," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 3688–3693.
- [6] M. Kubisch, H. Karl, A. Wolisz, L. C. Zhong, and J. Rabaey, "Distributed algorithms for transmission power control in wireless sensor networks," in *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, vol. 1. IEEE, 2003, pp. 558–563.
- [7] J. Deng, Y. S. Han, P.-N. Chen, and P. K. Varshney, "Optimal transmission range for wireless ad hoc networks based on energy efficiency," *IEEE Transactions on Communications*, vol. 55, no. 9, pp. 1772–1782, 2007.
- [8] M. Aboubakar, M. Kellil, A. Bouabdallah, and P. Roux, "Toward intelligent reconfiguration of rpl networks using supervised learning," in *2019 Wireless Days (WD)*. IEEE, 2019, pp. 1–4.
- [9] M. Meghji and D. Habibi, "Transmission power control in multihop wireless sensor networks," in *2011 Third International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2011, pp. 25–30.
- [10] K. Heurtefeux and H. Menour, "Experimental evaluation of a routing protocol for wireless sensor networks: Rpl under study," in *6th Joint IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE, 2013, pp. 1–4.
- [11] R. Zhang and J.-M. Gorce, "Optimal transmission range for minimum energy consumption in wireless sensor networks," in *2008 IEEE Wireless Communications and Networking Conference*. IEEE, 2008, pp. 757–762.
- [12] C. Song, M. Liu, J. Cao, Y. Zheng, H. Gong, and G. Chen, "Maximizing network lifetime based on transmission range adjustment in wireless sensor networks," *Computer Communications*, vol. 32, no. 11, pp. 1316–1325, 2009.
- [13] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5-6, pp. 183–197, 1991.
- [14] M. S. Rahman, Y. Park, and K.-D. Kim, "Localization of wireless sensor network using artificial neural network," in *2009 9th International Symposium on Communications and Information Technology*. IEEE, 2009, pp. 639–642.
- [15] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *First IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2006)*, 2006.
- [16] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in neural information processing systems*, 2011, pp. 2546–2554.
- [17] K. Will, "A conceptual explanation of bayesian hyperparameter optimization for machine learning," <https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f>, accessed: 2020-01-05.
- [18] "Metrics and scoring: quantifying the quality of predictions," https://scikit-learn.org/stable/modules/model_evaluation.html, accessed: 2020-01-05.
- [19] "Home - keras documentation," <https://keras.io/>, accessed: 2019-11-01.
- [20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.