COSIMIA : Combiner Simulation et Apprentissage Automatique pour Optimiser la Configuration des Réseaux IoT

Samir Si-Mohammed, Thomas Begin, Isabelle Guérin Lassous, Pascale Vicat-Blanc

▶ **To cite this version:**

Samir Si-Mohammed, Thomas Begin, Isabelle Guérin Lassous, Pascale Vicat-Blanc. COSIMIA : Combiner Simulation et Apprentissage Automatique pour Optimiser la Configuration des Réseaux IoT. CoRes 2023 - 8èmes Rencontres Francophones sur la Conception de protocoles, l'évaluation de performances et l'expérimentation de Réseaux de communication, May 2023, Cargese, France. hal-04086106

HAL Id: hal-04086106
https://hal.science/hal-04086106

Submitted on 1 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

COSIMIA : Combiner Simulation et Apprentissage Automatique pour Optimiser la Configuration des Réseaux IoT

Samir Si-Mohammed[1] et Thomas Begin[1] et Isabelle Guérin Lassous[1] et Pascale Vicat-Blanc[2]

[1] *ENS de Lyon, Université Claude Bernard Lyon 1, CNRS, LIP, France*
[2] *Inria, ENS de Lyon, Université Claude Bernard Lyon 1, CNRS, LIP, France*

L'internet des objets (IoT) couvre un large éventail d'applications qui s'appuient sur une ou plusieurs technologies réseau pour échanger des données entre les différents dispositifs IoT et les applications utilisateur. Ces technologies sont très hétérogènes et sont caractérisées par un ensemble de paramètres de configuration spécifiques à chacune. Les architectes IoT peuvent avoir recours à la simulation pour sélectionner une configuration réseau qui permette d'obtenir les meilleurs performances pour une application et un contexte cibles. Mais cette approche peut s'avérer coûteuse en temps de calcul et donc en énergie. Dans cet article, nous proposons COSIMIA, une méthode s'appuyant sur la combinaison de la Simulation et de l'Apprentissage automatique afin d'accélerer la recherche d'une configuration optimisée pour un scénario IoT donné. La méthode proposée est appliquée à deux études de cas inspirées de situations industrielles réelles, utilisant des technologies IoT différentes. Nous montrons que cette méthode permet de déterminer de bonnes configurations réseau avec un nombre réduit de simulations.

Mots-clefs : Internet des Objets, Simulation, Apprentissage Automatique, Optimisation, Évaluation de Performance.

1 Introduction

Les besoins de connectivité et de communication très hétérogènes des applications IoT a donné naissance à une importante profusion de technologies de réseaux, notamment à faible consommation d'énergie (LPWAN) permettant l'échange de données entre les différents dispositifs connectées et les applications utilisateur. De manière générale, chaque technologie IoT possède un ensemble de paramètres de configuration spécifiques pouvant impacter sa performance. Par exemple, le facteur d'étalement en LoRa (ou SF pour Spreading Factor) qui varie de 7 à 12, peut agir sur la portée de communication (*e.g.,* de 1 à 10 km), ainsi que sur le temps pour qu'un message arrive à destination (*e.g.,* de 70 ms à 2 secondes). Déterminer la valeur adéquate du SF à utiliser est donc une question importante pour l'architecte IoT lors du déploiement de sa solution. De plus, étant donnés le nombre de paramètres ainsi que leur valeurs respectives à définir, l'ensemble des combinaisons pour le paramètrage de LoRa devient très vite considérable. Enfin, tester ces combinaisons à travers un déploiement réel s'avère onéreux en argent et en temps. Ce constat est généralisable à de nombreuses technologies de communication.

C'est pourquoi il est fréquent d'avoir recours à la simulation. Même si cette approche peut s'avérer coûteuse en temps, elle offre une certaine facilité dans le test des configurations et dans l'évaluation des performances. Afin de diminuer le temps de simulation, nous proposons COSIMIA, qui s'appuie sur l'apprentissage automatique pour exploiter une partie seulement de l'espace de configurations. La principale contribution de cet article est de montrer comment l'apprentissage automatique peut accélèrer la simulation réseau et en réduire les coûts de calcul. Notre méthode combinant simulation et IA permet d'optimiser la configuration d'une technologie réseau pour un scénario IoT donné, avec un nombre de simulations réduit. En outre, nous présentons les résultats de son application à deux cas d'études réels inspirés de l'IoT. Les résultats montrent que la solution permet d'obtenir des configurations proches à 99% de la solution optimale, avec une réduction considérable du temps d'exécution, pouvant aller jusqu'à un facteur de 60.

2 État de l'art

Plusieurs travaux se sont intéressés à l'optimisation des configurations réseau des technologies IoT. Pour LoRaWAN par exemple, les auteurs de [Sal] proposent une approche de programmation linéaire pour trouver les valeurs optimales du Carrier Frequency et du Spreading Factor qui réduisent le nombre de collisions ainsi que la consommation d'énergie. [Ben] quant à eux proposent un modèle mathématique afin de trouver les valeurs du Spreading Factor, de la largeur du canal ainsi que de la puissance d'émission pour minimiser la consommation d'énergie. Pour ce qui est de 6LoWPAN, les paramètres MAC de la norme 802.15.4 sont souvent ceux que l'on cherche à optimiser. Par exemple, les auteurs de [Al-17] proposent d'utiliser un réseau de neurones pour trouver les valeurs optimales de ces paramètres MAC afin de maximiser le débit et de minimiser la latence. La solution COSIMIA diffère de ces méthodes dans le fait qu'elle soit complètement générique : Elle peut être appliquée à n'importe quelle technologie IoT, sur n'importe quels paramètres de configurations bornés et selon n'importe quelles métriques de performances. La seule condition est que la technologie, les paramètres et les métriques en question soient disponibles sur un simulateur réseau.

3 Description de la Méthode

3.1 Conception

Modélisation de l'application : Tout d'abord, il nous faut caractériser un scénario IoT. Pour ce faire, nous le définissons par : (i) Le nombre d'appareils (end-devices), (ii) la taille des messages (en octets), (iii) la périodicité des messages (en secondes), (iv) la portée du déploiement, qui représente la distance maximale pouvant séparer deux appareils (en mètres) et (v) l'environnement radio pouvant être (a) rural, (b) urbain, (c) suburbain ou (d) indoor. Nous considérons que ces cinq paramètres sont suffisants pour obtenir une caractérisation de première ordre d'un scénario IoT.

Configuration réseau : Comme mentionné auparavant, chaque technologie IoT est caractérisée par un ensemble de paramètres souvent propres à chaque technologie IoT. Nous considérons de plus le nombre de passerelles comme étant un paramètre commun à toutes les technologies (à topologie en étoile) et qu'il faut optimiser. Nous définissons une configuration réseau par une combinaison de ces paramètres (y compris donc le nombre de passerelles).

Simulation : La simulation réseau est un outil puissant permettant aux utilisateurs de simuler des déploiements réseau à moindre coût. L'objectif de la simulation réseau est de reproduire le comportement d'une technologie IoT sans avoir à utiliser du vrai matériel. COSIMIA s'appuie sur le simulateur ns-3.

KPIs (Key Performance Indicators) : Les KPIs sont des métriques permettant l'évaluation de la performance d'une technologie IoT pour un scénario IoT donné. Nous en considérons dans notre cas quatre : (i) Le taux de succès des messages, qui est le pourcetange des messages correctement reçus parmi tous ceux envoyés, (ii) la consommation énergétique qui représente la quantité d'énergie consommée par les appareils durant le déploiement, (iii) la latence qui est le temps moyen que prend un message pour transiter de la source vers la destination et (iv) le coût, qui dans notre cas représente le prix d'achat des passerelles. Une autre formule de calcul de coût pourrait bien entendu être envisagée.

Scoring : Un score est utilisé pour évaluer chaque configuration réseau, sur la base des KPIs obtenus lors de la simulation. Trouver la meilleure configuration réseau revient à trouver celle qui a le score le plus élevé. Une façon courante de comparer différentes alternatives en fonction de différents attributs, consiste à utiliser des méthodes MADM (Multi-Attribute Decision Making). Dans notre cas, les alternatives sont les configurations réseau, tandis que les attributs sont les KPIs considérés.

Régression : La régression est une branche de l'apprentissage automatique, où des algorithmes sont utilisés pour prédire des résultats continus à partir de caractéristiques d'entrée données. Dans notre cas, les entrées sont les configurations réseau, et la sortie est le score calculé. Ainsi, nous serons en mesure de déterminer le score d'une configuration réseau sans passer par la simulation.

3.2 Fonctionnement

COSIMIA se déroule en trois étapes. Nous les présentons dans ce qui suit.

1. Génération des données : Cette première étape consiste à échantillonner l'espace des configurations réseau. Pour chaque paramètre, à l'exception du nombre de passerelles, nous considérons la valeur minimale, moyenne et maximale (*e.g.,* les valeurs 7, 10 et 12 pour le SF en LoRa). En ce qui concerne le nombre

de passerelles, nous l'itérons de 1 à une valeur maximale que nous fixons au nombre de d'appareils divisé par 5, avec un pas de 2. Ensuite, ces différentes configurations réseau sont testées par simulation, en utilisant ns-3 dans notre cas. Les KPIs résultants sont utilisés pour attribuer un score à chaque configuration réseau, en utilisant TOPSIS (Technique for Order Preference Similarity to Ideal Solution) [Lai94] dans notre cas. À la fin de cette étape, nous avons un ensemble de données composé de différentes configurations réseau, où chacune a un score basé sur les KPIs résultants de la simulation.

2. Entraînement : La deuxième étape consiste à apprendre à des modèles de régression à faire la prédiction du score d'une configuration réseau donnée, sans utiliser de simulation. Pour ce faire, nous soumettons l'ensemble des échantillons générés durant l'étape précédente à différents modèles de régression (Gradien boosting, Random forest, KNN, SVR et Linear Regression), dont les variables d'entrée sont les paramètres de configuration du réseau et dont la sortie est le score. Ainsi, les modèles "apprennent" à prédire le score d'une configuration réseau donnée.

3. Recherche : Enfin, il reste à employer les modèles entraînés afin de trouver la meilleure configuration réseau. Pour ce faire, une recherche exhaustive est effectuée pour toutes les configurations réseau possibles. Pour chacune d'entre elles, au lieu d'exécuter la simulation, nous utilisons les modèles entraînés pour calculer le score. La recherche exhaustive est rendue possible par le fait que la prédiction utilisant les modèles de régression se fasse quasi immédiatement. La configuration réseau qui renvoie le meilleur score est celle qui est retenue.

4 Application

Nous appliquons COSIMIA sur 2 cas d'études : (i) Télémetrie en LoRaWAN et (ii) Smart watering en 6LoWPAN. Afin de valider les résultats de COSIMIA, nous effectuons des simulations de façon exhaustive sur l'ensemble des combinaisons possibles des paramètres afin d'obtenir la solution optimale. Nous comparons ensuite la solution retournée par chaque modèle de régression en calculant la "proximité", que l'on définit par le ratio du score de la solution retournée par le modèle sur le score de la solution optimale. Nous comparons également le temps nécessaire pour les deux approches.

4.1 Cas d'Étude A : Télémetrie en LoRaWAN

Pour ce premier cas d'étude, nous appliquons COSIMIA à un scénario de déploiement de télémetrie en LoRaWAN avec le scénario IoT suivant : 200 capteurs envoient aux passerelles des messages de 50 octets toutes les 600 secondes. La distance maximale séparant les appareils est de 8000 mètres, et le déploiement est en environment rural. Les paramètres considérés pour LoRaWAN sont les suivants : Le nombre de passerelles (nGW), le Spreading Factor (SF), pouvant prendre des valeurs entre 7 et 12, le Cyclic Redundancy Check (CRC) à 0 ou 1, le Coding Rate (CR) entre 1 et 4 et le type de trafic, avec 0 pour le trafic sans ACK, 1 pour le trafic confirmé (avec ACK). La Table 1 indique les résultats obtenus pour ce premier cas d'étude. La recherche exhaustive utilisant la simulation nous montre que la solution optimale comprend 5 passerelles, un SF de 8, un trafic non confirmé, un CR de 1 et un CRC de 0. La détermination de cette solution a nécessité 7,5 heures et 3840 simulations, tandis que COSIMIA a nécessité 1,16 heures et 480 simulations. Ceci équivaut donc à une diminution du temps de simulation d'un facteur de 6. Nous constatons que la plupart des modèles de régression atteignent 99% de proximité. La régression linéaire quant à elle peine à dépasser les 81% de proximité, ce qui corrobore le fait que le problème n'est pas linéaire.

Modèle de régression	Solution trouvée	KPIs				Génération des données		Proximité (%)
		Taux de Succès (%)	Consommation énergétique (Milli-Watts)	Latence (ms)	Coût ($)	Temps (minutes)	Nombre de simulations	
Exhaustive	[5,8,0,1,0]	99	0.082	195	5000	441	3840	N/A
Gradient boosting	[5,7,0,1,1]	89.5	0.093	112	500	70	480	99
Random forest	[5,7,0,1,1]	89.5	0.05	112	5000			99
KNN	[5,8,0,2,1]	99	0.082	195	5000			99.7
SVR	[10,7,0,1,1]	100	0.048	107	10000			94
Linear regression	[1,7,0,1,0]	4.45	0.047	107	1000			81

TABLE 1 : Résultats du cas d'étude A. Le format des solutions est le suivant : [nGW,SF,Type-Trafic,CR,CRC].

4.2 Cas d'Étude B : Smart watering en 6LoWPAN

Pour ce deuxième cas d'étude, nous considérons un scénario de smart watering employant 6LoWPAN (802.15.4) : 50 capteurs disposés dans un environnement sub-urbain envoient un message de 100 octets, indiquant le niveau d'eau, par seconde. La portée du déploiement est de 200 mètres. Les paramètres considérés sont : Le nombre de passerelles (nGW), Le nombre maximal de retransmission de trames (FR), entre 0 et 7, le nombre maximal de Backoffs (CB), entre 0 et 5, l'exposant maximal du Backoff (MaxBE) entre 3 et 8 et l'exposant minimal du Backoff (MinBE) entre 0 et 7. La Table 2 indique que l'amélioration en termes de temps d'exécution est nettement plus importante pour ce cas d'étude. En effet, nous passons de 22,7 heures à 0,42 heures, ce qui équivaut à une amélioration d'un facteur de 60. Ceci est dû au fait que les paramètres de configuration propres à 6LoWPAN ont en moyenne des cardinalités plus importantes que les paramètres de configuration propres à LoRaWAN. Toutefois, la proximité reste de l'ordre de 99% pour la plupart des modèles, avec toujours la même contre-performance du modèle de régression linéaire.

Modèle de régression	Solution trouvée	KPIs				Génération des données		Proximité (%)
		Taux de Succès (%)	Consommation énergétique (Watts)	Latence (ms)	Coût ($)	Temps (minutes)	Nombre de simulations	
Exhaustive	[3,4,3,4,0]	92	0.03	5.56	300	1367	23040	N/A
Gradient boosting	[3,5,3,5,3]	99.37	0.032	5.58	300	26	405	99
Random forest	[3,8,7,5,3]	100	0.033	31.16	300			98
KNN	[3,8,7,5,6]	100	0.033	31.16	300			98
SVR	[5,7,7,5,6]	100	0.03	30.15	500			94
Linear regression	[10,8,7,5,7]	100	0.02	26.02	1000			79

TABLE 2 : Résultats du cas d'étude B. Le format des solutions est le suivant : [nGW,MaxBE,MinBE,CB,FR].

4.2.1 Impact de la Granularité de l'Échantillonage

Nous montrons dans ce qui suit l'impact de la granularité de l'échantillonage dans l'étape de génération des données sur les performances de la méthode. La proximité moyenne est calculée à partir des proximités de tous les modèles de régression testés.

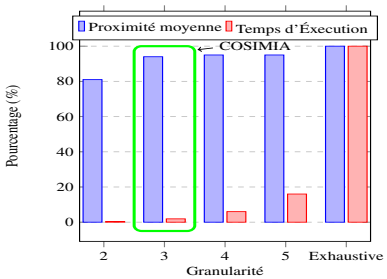

FIGURE 1 : Impact de la granularité de l'échantillonage sur les performances.

Naturellement, la Figure 1 montre que plus la granularité est fine (en d'autres termes, plus on prend de points pour l'échantillonage) et plus le temps d'exécution croît. Elle montre aussi qu'à partir d'une granularité de 3 (valeurs minimale, intermédiaire et maximale comme nous le préconisons dans COSIMIA), la proximité moyenne atteint 95% et reste fixe à cette valeur pour les granularités supérieures.

5 Conclusion

Nous avons présenté une méthode s'appuyant sur la simulation ainsi que sur l'apprentissage automatique pour optimiser la configuration réseau d'une technologie IoT dans un contexte donné. La méthode se déroule en trois étapes : (i) La génération des données faite à l'aide d'un échantillonage, de simulation et d'une fonction de scoring, (ii) l'apprentissage où la fonction de scoring est apprise en fonction des configurations réseau et (iii) une recherche s'appuyant sur les modèles de régression entraînés. Nous avons illustré l'application sur deux cas d'études comprenant des scénarios et technologies IoT différentes. Nos résultats montrent la capacité de la méthode à retourner des solutions proches à 99% de la solution optimale, et ce avec une diminution importante du temps d'exécution. Dans des travaux futurs, nous comptons comparer notre solution à d'autres solution existantes pour l'optimisation de configurations réseau.

Références

[Al-17] Al-Kaseem, Bilal et al. A new intelligent approach for optimizing 6lowpan mac layer parameters. *IEEE Access*, 2017.

[Ben] Benatti, Rafael Senna et al. An optimization method based on lora parameters for energy consumption reduction. In *2021 IEEE International Symposium on Instrumentation Systems, Circuits and Transducers*.

[Lai94] Lai, Young-Jou et al. TOPSIS for MODM. *European journal of operational research*, 1994.

[Sal] Sallum, Eduardo et al. Performance optimization on lora networks through assigning radio parameters. In *2020 IEEE International Conference on Industrial Technology (ICIT)*.