



HAL
open science

Validation d'algorithmes coopératifs dans les réseaux de véhicules

Guillaume Béduneau, Ghada Jaber, Bertrand Ducourthial

► **To cite this version:**

Guillaume Béduneau, Ghada Jaber, Bertrand Ducourthial. Validation d'algorithmes coopératifs dans les réseaux de véhicules. 24èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (ALGOTEL 2022), May 2022, Saint-Rémy-Lès-Chevreuse, France. hal-03656246

HAL Id: hal-03656246

<https://hal.science/hal-03656246>

Submitted on 2 May 2022

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.

Validation d'algorithmes coopératifs dans les réseaux de véhicules

Guillaume Béduneau and Ghada Jaber and Bertrand Ducourthial[†]

Université de technologie de Compiègne, CNRS, Heudiasyc (Heuristique et Diagnostic des Systèmes Complexes), CS 60 319 - 60 203 Compiègne Cedex

Dans un système de transport intelligent, la coopération entre véhicules est assurée par des algorithmes répartis basés sur des échanges entre véhicules proches. Le déploiement de ces algorithmes de coopération nécessite leur validation. Mais prédire le comportement d'un algorithme réparti dans un réseau dynamique est une tâche complexe, parfois coûteuse pour un résultat pas toujours réaliste. Nous proposons une méthode de prédiction du comportement d'un algorithme selon la dynamique du réseau où il s'exécute.

Keywords: validation d'algorithme, réseau véhiculaire, réseau dynamique, ITS, VANET

1 Introduction

Problème. Avec l'avènement des systèmes de transport intelligents, la coopération entre véhicules pourrait permettre de résoudre à la fois les problèmes de sécurité routière et de trafic congestionné. Pour des raisons légales, les algorithmes répartis assurant cette coopération ont besoin de démontrer leur efficacité à résoudre les problèmes ciblés. En raison de la nature dynamique des réseaux véhiculaires, démontrer l'efficacité ou la fiabilité d'un algorithme reste une tâche difficile.

Dans ce contexte, la validation d'une application véhiculaire est à la fois difficile et nécessaire avant tout déploiement. Démontrer l'algorithme implémenté par l'application offre des garanties de fonctionnement, mais cela oblige généralement à faire des hypothèses qui pourraient impacter le réalisme de ces garanties. Tester l'application implémentée avec des véhicules expérimentaux en expérience réelle est très coûteux (financièrement et logistiquement). En modélisant une partie du réel (les communications en émulation, la plupart des composants en exécution réelle), on s'abstrait de ces contraintes mais il reste difficile d'extraire de telles expériences des garanties générales rassurant les acteurs sur le bon fonctionnement de l'application dans des circonstances non testées.

Contribution. Pour résoudre ce problème, nous proposons de dégager les propriétés désirées d'une exécution de l'algorithme et de les exprimer grâce à une modélisation adéquate du réseau dynamique. La vérification des propriétés du réseau dynamique suffit alors à prédire si l'algorithme remplira les propriétés désirées. Le choix de la modélisation utilisée contraint les propriétés exprimables et donc les algorithmes étudiés. Nous utilisons le modèle des p -graphes dynamiques, qui permet de s'abstraire du temps. Un p -graphe dynamique est une séquence de graphes dont chaque arête permet de transporter p messages avant de disparaître.

Dans cet article, nous décrivons notre démarche et des expériences menées pour sa validation. Nous montrons qu'il est possible de prédire avec une certaine précision les performances d'applications réparties pour divers scénarios routiers.

2 État de l'art

Graphes dynamique. Les réseaux dynamiques ont d'abord été étudiés pour leurs propriétés structurelles [1, 8]. La notion de *evolving graph* proposée par [3, 7] permet d'étudier les évolutions dans le temps de

[†]Ce travail bénéficie du soutien de l'AID et du Labex MS2T (Investissements d'avenir) sous le numéro ANR-11-IDEX-0004-02.

la topologie dans une séquence de graphes. Par exemple, la *journey* est un chemin dans la séquence de topologies successives. Les *temporal reachability graphs* sont définis dans [11] en considérant une arête dès lors qu'une *journey* existe entre les extrémités.

Dans [4], des calculs répartis locaux basés sur des renumérotations de graphes et des *evolving graphs* sont utilisés pour analyser et comparer des algorithmes répartis. Dans [5], une synthèse est proposée avec un modèle unifié dénommé *Time-Varying Graph* (TVG). Plusieurs classes de TVG sont identifiées et l'impact de leurs propriétés sur les algorithmes répartis est étudié. Dans [10], les TVG sont modélisés avec un quadruplet $G_d = (V, E, T, \Phi)$ qui inclut notamment les dates de début et de fin de chaque arête. Dans [9], les auteurs modélisent un réseau dynamique via une agrégation par fenêtres temporelles des liens dynamiques datés. Ils montrent qu'il est possible d'étudier la propagation dans un réseau dynamique avec des fenêtres temporelles de plus en plus larges, jusqu'à un seuil calculable à partir duquel la modélisation ne rend plus compte des propriétés de propagation.

Modèle des p -graphes dynamiques. Dans [6], la modélisation du réseau dynamique s'abstrait de la notion de temps en introduisant les *p -graphes dynamiques*; elle semble mieux appropriée pour les preuves des algorithmes répartis. Les travaux résumés dans cet article en sont la continuité. Dans un p -graphe, chaque arête (appelée p -arête) représente une situation où les deux nœuds adjacents peuvent échanger p messages au moins. Une topologie de réseau dynamique réelle (appelée DNO pour *Dynamic Network Observation*) peut être représentée par une suite de p -graphes (pour p fixé). En fusionnant deux p -graphes consécutifs identiques, nous obtenons le p -graphe dynamique de la DNO. Itérer ce procédé pour chaque entier non nul p , donne la *famille de p -graphes dynamiques* représentant la DNO.

3 Méthode proposée

Dans cette section, nous décrivons notre méthodologie de prédiction du comportement des algorithmes répartis dans les réseaux véhiculaires.

Propriété de performance. Lorsqu'on prouve un algorithme réparti, on démontre que si les hypothèses de la preuve sont réunies, les spécifications de l'algorithme seront remplies. Dans les faits, il est souvent attendu que la prédiction aille plus loin qu'un résultat binaire, en raison des contraintes sur les ressources utilisées; c'est donc une prédiction de performance qui est obtenue.

On peut toujours exprimer la performance d'un algorithme à travers un prédicat indiquant si les performances qu'il atteint sont satisfaisantes. Un tel prédicat peut être adapté finement au cas d'usage dans lequel l'algorithme est étudié. Il est appelé dans ce travail *Propriété de Performance* et noté *PP*.

Description de la méthode. Partant d'un problème donné, l'étude de l'algorithme qui le résout en réseau dynamique permet de dégager une propriété de performance dont la réalisation dépend de la dynamique du réseau. En exprimant cette propriété sous forme de condition sur une famille de p -graphes dynamiques, il est possible de vérifier la véracité de cette propriété avec uniquement les DNO (représentatives des cas d'usages étudiés). En effet, on peut facilement calculer la famille de p -graphes dynamiques représentant la DNO puis vérifier automatiquement si la PP y est satisfaite.

Implémentation proposée. La propriété de performance est obtenue à partir d'une analyse du problème et de l'algorithme. Leur étude permet ensuite l'expression de la PP sous forme de condition sur la famille de p -graphes.

À partir de traces GPS représentatives d'un trafic routier donné et de la portée des communications sans fil, on peut calculer la DNO correspondante. Nous utilisons pour cela l'émulateur de la suite Airplug [2] qui permet de rejouer des expériences routières en variant les paramètres.

La construction de la famille des p -graphes dynamiques est réalisée par un programme externe nommé *pgraphtool* qui traite les données de l'émulateur. La propriété de performance est détectée via des expressions régulières appliquées sur la sortie de *pgraphtool*.

Résultat prédit	PP réalisée	PP réalisée	PP non réalisée	PP non réalisée
Vérité terrain	PP réalisée	PP non réalisée	PP réalisée	PP non réalisée
Exactitude de prédiction	Succès prédit	Échec non prédit	Succès non prédit	Échec prédit

TABLE 1: Exactitude de la prédiction selon les valeurs de la vérité terrain et de la prédiction.

4 Validation expérimentale.

Dans cette section, nous décrivons certaines des expériences menées afin de mettre à l'épreuve la méthode de prédiction proposée. D'autres expériences mettant en jeu des algorithmes différents ont également donné d'excellents résultats.

Protocole expérimental. Valider une méthode de prédiction de comportement algorithmique nécessite une « vérité terrain », obtenue à partir de l'exécution réelle de cet algorithme dans les mêmes conditions.

Nous implémentons donc un algorithme réparti dans une application compatible avec la suite Airplug et l'émulateur exécute l'application dans le scénario expérimental. Au cours de l'exécution, chaque instance de l'application génère des fichiers de log retraçant son comportement lors de l'exécution. On peut analyser ces fichiers de log pour vérifier si l'algorithme a atteint des performances satisfaisantes (la propriété de performance est réalisée), ce qui constitue la vérité terrain.

En utilisant la méthode décrite dans la section 3, on prédit le résultat à partir uniquement de la famille de p -graphes représentant la DNO issue des traces GPS sans tenir compte de l'application répartie ni de son exécution. On compare les deux résultats (le résultat prédit et la vérité terrain) pour obtenir l'exactitude de la prédiction comme présenté à la table 1.

Algorithme. La validation a été conduite sur plusieurs algorithmes répartis en contexte véhiculaire, incluant l'algorithme de diffusion avec ou sans retour et un algorithme de beaconing. Nous présentons ici les résultats concernant un algorithme nommé n -ACK de transfert de données avec acquittement.

Dans cet algorithme, un nœud source transmet une donnée fragmentée à un nœud de destination. Le nœud de destination envoie un accusé de réception cumulatif tous les n fragments reçus, qui permet à la source d'adapter les envois en prenant en compte les éventuels fragments manqués par la destination. Lorsque la destination ne reçoit plus de messages de la source, elle devient source avec les fragments qu'elle a reçus.

Scénarios. Le cas d'usage présenté dans ce travail consiste en une borne de bord de route b disposant d'une donnée fragmentée en 24 fragments. Elle envoie les fragments aux véhicules approchant en utilisant l'algorithme n -ACK.

Un camion c et une voiture v circulent sur l'autoroute à proximité de b pendant que v , ayant une vitesse supérieure, dépasse c . Si v ne reçoit pas tous les fragments de b , il peut en récupérer de c lorsque c deviendra source après s'être éloigné de b .

Dans ce contexte, la propriété de performance PP exprime le fait que v a bien reçu toute la donnée (24 fragments différents). Exprimée selon la famille de p -graphes dynamiques, on obtient le résultat présenté à l'équation 1. Dans cette équation, si e est une p -arête, $C(e)$ est sa *capacité*, c'est-à-dire la valeur maximale de p telle que e apparaît dans le p -graphe dynamique.

$$PP_{n\text{-ACK}} \equiv \min \left(\sum_{\substack{e : n\text{-edge} \\ (b_1, c_1)}} C(e), \sum_{\substack{e : n\text{-edge} \\ (c_1, v_1)}} C(e) \right) + \sum_{\substack{e : n\text{-edge} \\ (b_1, v_1)}} C(e) \geq N \quad (1)$$

Résultats. Les résultats des tests menés sur le dépassement autoroutier avec l'algorithme de transfert n -ACK sont présentés en figure 1. On peut remarquer que les erreurs de prédiction sont extrêmement faibles (moins de 10%). Pour des raisons de sécurité routière, il peut être utile de privilégier les succès non prédits aux échecs non prédits (qui sont dus ici au fait que la propriété de performance n'est pas suffisante pour assurer le succès de l'algorithme). On pourra alors dériver la propriété de performances d'une preuve algorithmique (ou autre condition suffisante). Il est également possible d'adapter la propriété de performance

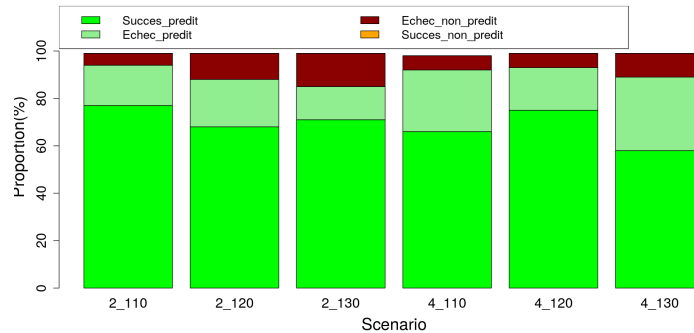


FIGURE 1: Exactitude des prédictions réalisées sur l’algorithme n -ACK dans le cadre d’un dépassement autoroutier. Sous chaque barre, le premier chiffre est la valeur de n et le nombre suivant est la vitesse de la voiture v_1 (km/h).

PP en y ajoutant une marge de sécurité (ici, cela pourrait être d’effectuer la prédiction en considérant que la donnée fait 25 fragments).

5 Conclusion

La conception des algorithmes répartis destinés aux réseaux dynamiques est complexe. Leur validation par des preuves est généralement peu soucieuse du réalisme des hypothèses tandis que les validations par tests en simulateur peuvent manquer de précision pour fournir des garanties de sécurité.

Grâce à une étude algorithmique associée aux p -graphes dynamiques, nous caractérisons la dynamique des réseaux, afin de proposer une méthodologie de prédiction du comportement des algorithmes dans les réseaux véhiculaires. L’évaluation de cette méthodologie sur des cas réalistes donne des résultats prometteurs, dont certains sont présentés dans cet article. Dans la suite, nous envisageons d’améliorer nos prédictions afin d’éprouver notre approche sur des applications et des scénarios plus complexes (pertes aléatoires par exemple). Des expériences réelles d’application de la méthode sur des cas d’usage réalistes (protection des piétons) sont également en cours de préparation.

Références

- [1] P. Borgnat, E. Fleury, J.-L. Guillaume, and C. Robardet. Characteristics of the dynamic of mobile networks. In *Proc. BIONETICS*, vol. 6811, pages 130–139, 2009.
- [2] A. Buisset, B. Ducourthial, F. El Ali and S. Khalfallah. Vehicular networks emulation. *ICCCN 2010*.
- [3] B.-M. Bui-Xuan, A. Ferreira, and A. Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. In *journal IJFCS*, 14(2) :267–285, 2003.
- [4] A. Casteigts, S. Chaumette, and A. Ferreira. Distributed computing in dynamic networks : Towards a framework for automated analysis of algorithms. *CoRR*, abs/1102.5529, 2012.
- [5] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. In *Proc. ADHOC-NOW 2011*, pp. 346–359.
- [6] B. Ducourthial, A. Wade. Dynamic p -graphs for capturing the dynamics of distributed systems. *Ad Hoc Networks*, vol. 50, pp. 13-22, 2016.
- [7] A. Ferreira. Building a reference combinatorial model for MANETs. *IEEE Network*, 18(5) :24–29, 2004.
- [8] F. Harary and G. Gupta. Dynamic graph models. *Mathematical and Computer Modelling*, 25(7) :79 – 87, 1997.
- [9] Y. Léo and C. Crespelle and E. Fleury. Non-altering time scales for aggregation of dynamic networks into series of graphs. In *Computer Networks 2019*.
- [10] K. Wehmuth, É. Fleury and A. Ziviani. A New Model for Time-Varying Graphs. In *Temporal and Dynamic Networks : From Data to Models*, , March 2013.
- [11] J. Whitbeck, M. Dias de Amorim, V. Conan, and J.-L. Guillaume. Temporal reachability graphs. *Mobicom 2012*.