ZebraNet Analysé dans le Modèle des Protocoles de Population

Joffroy Beauquier [(1)], Janna Burman [(2) †] et Valentin Malykh [(3)]
[(1)] *LRI, UMR 8623, University Paris Sud and INRIA Saclay, Orsay 91405, France*
[(2)] *INRIA, MASCOTTE Project, 2004, route des Lucioles, B.P. 93, F-06902 Sophia-Antipolis Cedex*
[(3)] *Moscow Institute of Physics and Technology (State University)*
† Support d'une bourse Chateaubriand du Gouvernement Français

Résumé
Nous étudions le protocole de collecte de données du projet ZebraNet, dans le modèle des *protocoles de population*. Dans ce projet des capteurs sont attachés à une population de zèbres, en Afrique Centrale, et fournissent des données aux biologistes qui étudient leurs structures migratoires et comportementales. Nous montrons qu'un protocole voisin de celui utilisé dans ce projet ne se termine pas. Cela entraîne que le protocole originel ne se termine pas non plus. Aussi proposons nous une modification qui fournit la terminaison. Nous prouvons la correction de ce protocole modifié et nous analysons sa complexité en temps au pire, dans le modèle des protocoles de population avec *temps de couverture*. La comparaison de cette complexité avec celle du protocole optimal est très défavorable.
Le protocole de collecte de données de ZebraNet a fait l'objet de simulations, mais c'est la première fois, à notre connaissance, qu'est réalisée une étude purement analytique.

Mots-clé: réseaux de capteurs mobiles, protocoles de population, temps de couverture, algorithmes, analyse de complexité en temps

1 Introduction

Le modèle des protocoles de population (PP) a été introduit en 2004 [AADFP04], pour représenter des réseaux de capteurs mobiles très simples. Dans ce modèle, des agents mobiles anonymes se déplacent de manière asynchrone et peuvent communiquer et échanger des informations deux à deux, quand ils sont choisis par un ordonnanceur (ce qui peut être interprété en disant que leur mouvement les amène près l'un de l'autre). Un des buts des PP était de déterminer ce qui peut être calculé dans un réseau de capteurs mobiles avec des hypothèses minimales. C'est la raison pour laquelle les agents sont asynchrones, anonymes, disposent d'une mémoire petite et bornée et qu'aucune hypothèse n'est faite sur la manière dont ils se déplacent, à part une hypothèse d'équité. Comme la famille d'applications qui peuvent être calculées dans un tel modèle est plutôt limitée [AAER07], diverses extensions ont été suggérées. Parmi elles, citons la considération d'un agent distingué, avec des ressources étendues, la station de base [BCMRR07] ou des identificateurs uniques [GR07, BBK09]. [BBCK10] ajoute au modèle originel une *notion de vitesse* des agents. Cela s'exprime comme une contrainte sur l'ordonnanceur qui doit faire en sorte qu'un agent rencontre chaque autre agent dans un délai borné (son *temps de couverture*). Bien entendu, étant donné qu'il n'existe pas de temps réel dans de tels systèmes, la durée doit être comptée en termes d'*événements* (rencontres) globaux. Les temps de couverture permettent une approche quantitative des protocoles de population, en ce sens qu'ils permettent des analyses de complexité en temps (données en termes de temps de couverture). Le modèle originel ne le permettait pas.
La notion de temps de couverture est validée par certains jeux de données expérimentaux, comme ceux concernant des populations d'étudiants à l'UCSD ou à Dartmouth College, des visiteurs à Disneyland ou des participants à la conférence INFOCOM. L'étude statistique de ces jeux de données indique en effet que l'ICT (Inter Contact Time), considéré comme une variable aléatoire mesurant le temps entre les rencontres successives de deux agents, suit une loi de Pareto tronquée, ce qui implique en pratique que l'ICT est borné [RSHLC08]. Le temps de couverture d'un agent, étant directement corrélé au maximum des ICT de l'agent (avec les autres agents), est donc lui aussi borné.
Dans cet article nous proposons une analyse d'un protocole de collecte de données voisin de celui du projet ZebraNet [JOWMPR02], développé en Afrique Centrale par l'Université de Princeton. Ce projet utilise un protocole basé sur l'histoire passée, pour collecter les données à une station de base. Lorsqu'il a la possibilité de transmettre ses données à plusieurs agents (les capteurs attachés aux zèbres), un agent choisit celui qu'il estime avoir le plus de chances de rencontrer la station de base dans un futur proche. Cette estimation est basée sur le nombre de rencontres passées de cet agent avec la station de base.
Le premier résultat de notre étude est que ce protocole ne se termine pas. Il existe en effet des exécutions infinies dans lesquelles certaines valeurs n'atteignent jamais la station de base. Cette propriété n'avait pas été décelée par les simulations. Nous montrons cependant qu'une très légère modification du protocole (en fait l'ajout d'un bit à la mémoire de chaque agent) permet d'obtenir la terminaison sans changer fondamentalement la structure des exécutions (Sec. 2).

Nous nous intéressons alors à ce protocole modifié, dont nous prouvons la correction et pour lequel nous menons une étude de complexité (Sec. 3). Le résultat n'est pas bon, et le protocole est surclassé par le protocole présenté dans [BBCK10]. Celui-ci suppose cependant que lorsque deux agents se rencontrent, ils sont capables de déterminer lequel a le plus faible temps de couverture.
Nous ne donnons ci-dessous que quelques définitions et notations et nous renvoyons à [BBCK10] pour les définitions détaillées du modèle. Nous notons **A** l'ensemble des agents et **A*** l'ensemble de tous les agents moins la station de base (BS). Nous utilisons le terme *agents mobiles* pour les éléments de **A***. Le nombre d'agents, $|\mathbf{A}| = \mathbf{n}$, n'est pas connu des agents. La communication entre agents est symétrique. Lorsque a rencontre b, lors de l'événement (a, b), les états de a et de b peuvent être modifiés. Un système est donné avec un *vecteur de temps de couverture*, ayant pour dimension le nombre d'agents. Chaque exécution satisfait la *propriété de couverture*: pour tout agent a de temps de couverture $\mathbf{cv_a}$, dans tout segment d'exécution de longueur $\mathbf{cv_a}$ événements, l'agent a rencontre au moins une fois chacun des autres agents. Un vecteur de temps de couverture étant donné, nous notons $\mathbf{cv_{max}}$ et $\mathbf{cv_{min}}$ respectivement l'élément maximum et minimum de ce vecteur.

2 Les protocoles

Nous présentons une version simplifiée du protocole de ZebraNet (notée ZP), dans laquelle un agent n'a la possibilité de communiquer qu'avec un seul agent à la fois. Étant donné que les exécutions de ce protocole simplifié sont aussi des exécutions du protocole originel, la propriété de non terminaison se transporte. Pour des raisons de simplicité, nous supposons que lorsque deux agents mobiles se rencontrent, ils ont suffisamment de place dans leur mémoire pour recevoir les valeurs de l'autre (comme dans le projet ZebraNet). Nous dirons que l'un *transfère* ses valeurs à l'autre (aucune copie n'est conservée). *Accumulation* et *éloignement* sont des variables entières, initialisées à 0, et ***déclin*** est une constante entière.
Chaque fois qu'un agent rencontre BS, son *accumulation* est incrémentée et son *éloignement* est remis à 0. Chaque fois qu'un agent rencontre un autre agent mobile, différent de BS, *éloignement* est incrémenté. Si *éloignement* atteint la valeur de ***déclin***, *accumulation* est décrémentée. Lorsqu'un agent x transportant des valeurs rencontre un agent y, si la valeur d'*accumulation* de y est strictement plus grande que la valeur d'*accumulation* de x, x transfère toutes ses valeurs à y. L'agent x transfère systématiquement toutes ses valeurs lorsqu'il rencontre BS. Plus formellement:

<u>Lorsque l'agent x rencontre BS:</u>
1 <x transfère toutes les valeurs qu'il transporte à BS>;
2 $\mathit{accumulation}_x := \mathit{accumulation}_x + 1$;
3 $\mathit{éloignement}_x := 0$;

<u>Lorsque l'agent x rencontre y différent de BS:</u>
4 si ($\mathit{accumulation}_y > \mathit{accumulation}_x$) alors
5   <x transfère toutes les valeurs qu'il transporte à y>;
6 $\mathit{éloignement}_x := \mathit{éloignement}_x + 1$;
7 si ($\mathit{éloignement}_x >$ ***déclin*** et $\mathit{accumulation}_x \neq 0$) alors
8     $\mathit{accumulation}_x := \mathit{accumulation}_x - 1$;
9     $\mathit{éloignement}_x := 0$;

Remarque 1. ZP ne se termine pas.

Il n'est pas difficile de construire une exécution du protocole dans laquelle certaines valeurs n'atteignent jamais la station de base. Ceci est réalisé parce que certaines valeurs peuvent cycler entre deux agents, a et b. Chaque fois que a rencontre la station de base, les valeurs sont portées par b et chaque fois que b rencontre la station de base, elles sont portées par a. Bien entendu, les transferts des valeurs entre a et b sont rendus possibles par une gestion adéquate de leurs variables *accumulation*. Une telle exécution est aussi une exécution du protocole originel, qui ne se termine donc pas non plus.
La modification que nous proposons empêche qu'un agent qui a transmis une valeur à un moment donné puisse en recevoir ultérieurement. Une valeur ne peut donc pas cycler entre plusieurs agents, ce qui implique la terminaison puisque tout agent rencontre BS. Pratiquement, il suffit d'ajouter un bit d'activité *actif*. Initialisé à ***vrai***, il passe à ***faux*** quand l'agent mobile considéré effectue un transfert de valeurs vers un autre agent mobile. Une fois à ***faux*** le bit ne redevient jamais égal à ***vrai***. Dans le code, il suffit de modifier les lignes 4 et 5.
Nous notons MZP (Modified ZebraNet Protocol) le protocole modifié.

4' si ($accumulation_y > accumulation_x$ et $actif_y$ et <x transporte des valeurs>) alors
5' <x transfère toutes les valeurs qu'il transporte à y>; $actif_x$:= ***faux***;

3 Preuve de correction et analyse de complexité

Théorème 1. MZP se termine.

Preuve. Soit *ACT* l'ensemble des processus actifs (c'est-à-dire tels qu'*actif* soit ***vrai***) dans une configuration. Au cours d'une exécution, *ACT* est non croissant. Il existe donc un point de l'exécution à partir duquel *ACT* est constant. Nous affirmons alors que toutes les valeurs sont soit à la station de base, soit tenues par un agent actif. En effet, un agent non-actif ne peut tenir de valeur (la mémoire des agents permet le transfert de toutes les valeurs tenues). A partir du point où *ACT* est constant, il ne peut plus y avoir de transfert de valeurs entre processus actifs (cela ferait diminuer *ACT*), et il ne peut jamais y avoir de transfert de valeurs d'un processus actif à un processus non-actif (ligne 4'). Comme chaque agent actif rencontre BS tous les $\mathbf{cv}_{\mathbf{max}}$ événements, toutes les valeurs arrivent à BS au bout d'un nombre fini d'événements.

Nous présentons maintenant une étude analytique de complexité de MZP. Plus précisément, nous donnons des bornes, fonctions des temps de couverture, à la complexité au pire. L'étude utilise une suite de lemmes.

Lemme 1. Soit $e = e_1 e'$ une exécution de MZP, telle que la longueur de e_1 est $\mathbf{cv}_{\mathbf{max}}$. Nous notons C la configuration après e_1. Alors il existe un agent mobile a tel que, dans C, a est non-actif ou (non exclusif) toutes les valeurs initiales sont à la station de base.

Preuve. Supposons le contraire, c'est-à-dire que dans C tous les agents mobiles sont actifs et il existe une valeur qui n'est pas à la station de base. Puisque tous les agents mobiles sont actifs, aucune valeur n'a été transférée d'un agent mobile à un autre (ligne 5') et, puisque chaque agent a rencontré BS durant e_1 (la longueur de e_1 est $\mathbf{cv}_{\mathbf{max}}$) toutes les valeurs sont à la station de base. D'où une contradiction qui prouve le lemme.

Remarque 2. Soit a l'agent éventuellement mis en évidence par le lemme 1. Dans les mêmes conditions, après seulement $\mathbf{cv}_{\mathbf{a}}$ événements après la fin de e_1, a est non-actif.

Le lemme 1 sert de base à une induction sur le nombre k d'agents (k dans $\{1, 2, \dots, \mathbf{n}\text{-}2\}$), pour le cas $k = 1$.
Hypothèse d'induction: pour toute exécution $e = e_k e'$ de MZP, où e_k est de longueur $k \times \mathbf{cv}_{\mathbf{max}}$, on a la propriété suivante. Dans la configuration C_k après e_k, il existe un ensemble d'agents mobiles A_k, de cardinal k, différent de $\mathbf{A^*}$, tel que tout élément de A_k est non-actif dans C_k, ou (non exclusif) toutes les valeurs sont à la station de base.

Lemme 2. Supposons l'hypothèse d'induction vraie au rang k, $0 < k < \mathbf{n}$-1, et soit $e = e_{k+1} e'$ une exécution de MZP, où e_{k+1} est de longueur $(k+1) \times \mathbf{cv}_{\mathbf{max}}$. Nous notons C_{k+1} la configuration après e_{k+1}. Alors il existe un agent mobile a non dans A_k, tel que, dans C_{k+1}, a est non-actif ou (non exclusif) toutes les valeurs sont à la station de base. Si k est strictement inférieur à $\mathbf{n}$-2, nous posons $A_{k+1} = A_k \cup \{a\}$ pour poursuivre l'induction.

Preuve. Supposons le contraire, c'est-à-dire que tous les agents mobiles qui ne sont pas dans A_k sont actifs dans C_{k+1} et il existe une valeur qui n'est pas à la station de base. Soit a un agent tenant une telle valeur. a est nécessairement actif dans C_{k+1}. Soit C_k la configuration après $k \times \mathbf{cv}_{\mathbf{max}}$ événements. Tous les agents non dans A_k étaient actifs dans C_k, car un agent non-actif ne redevient jamais actif. Durant les $\mathbf{cv}_{\mathbf{max}}$ derniers événements de e_{k+1}, aucune valeur n'a été transférée d'un agent mobile à un autre et puisque chaque agent a rencontré BS toutes les valeurs sont à la station de base dans C_{k+1}. D'où une contradiction qui prouve le lemme.

Remarque 3. Soit a l'agent éventuellement mis en évidence par le lemme 2. Dans les mêmes conditions, après seulement $\mathbf{cv}_{\mathbf{a}}$ événements après avoir atteint C_k, a est non-actif.

Ce raisonnement fournit une borne supérieure à la longueur d'une exécution de MZP. Au bout de $(\mathbf{n}\text{-}1) \times \mathbf{cv}_{\mathbf{max}}$ événements, tout agent mobile est non-actif ou (non exclusif) toutes les valeurs sont à la station de base. Puisque la mémoire d'un agent lui permet de recevoir, le cas échéant, toutes les valeurs à transférer, les agents non-actifs ne peuvent tenir des valeurs et, dans les deux cas, toutes les valeurs sont à la station de base.

Théorème 2. La complexité au pire de MZP est bornée supérieurement par $(\mathbf{n}-1)\times\mathbf{cv}_{\mathbf{max}}$ événements.

Cette borne n'est pas très précise, car elle ne prend pas en compte toutes les contraintes liées à la propriété de couverture. Le dernier segment de longueur $\mathbf{cv}_{\mathbf{max}}$ est en effet raccourci car la propriété de couverture force la dernière rencontre avec BS à être plus précoce. Nous illustrons ce phénomène sur l'exemple ci-dessous.

Pour des raisons de simplicité, nous supposons dans l'exemple que tous les temps de couverture sont égaux à un même cv (ici $cv= 17$). Nous considérons $\mathbf{n} = 5$, de sorte que le résultat de borne supérieure donne $4\times cv$. Nous montrons qu'en fait nous pouvons construire une exécution de plus de $3\times cv$ événements. Le mécanisme de la construction peut être étendu, ce qui fournit une borne inférieure de $(\mathbf{n}-2)\times cv$, assez proche de la borne supérieure $(\mathbf{n}-1)\times cv$ sous l'hypothèse considérée.

Lorsque l'événement (a, b) provoque un transfert de valeurs de a vers b, nous le notons $(a, \underline{b})$. Le lecteur pourra vérifier que les transferts (ou l'absence de transfert) indiqués ci-dessous sont conformes aux valeurs des variables *accumulation*. Par exemple, lors du tout premier événement, (a_1, a_2), il n'y a pas de transfert parce que les variables *accumulation* de a_1 et de a_2 sont égales. Par contre, le premier événement $(a_3, \underline{a_1})$ provoque un transfert de a_3 vers a_1 car la variable *accumulation* de a_1 vaut 1 alors que celle de a_3 vaut 0. Chaque ligne, sauf la dernière correspond à un segment de longueur cv. On remarquera que la propriété de couverture est satisfaite sur l'exemple.

$(a_1, a_2)(a_1, a_3)(a_1, a_4)(a_2, a_3)(a_2, a_4)(a_3, a_4)(a_2, \underline{\mathrm{BS}})(a_4, \underline{\mathrm{BS}})(a_1, \underline{\mathrm{BS}})(a_3, \underline{a_1})(a_3, \mathrm{BS})(a_1, a_2)(a_1, a_3)(a_1, a_4)(a_2, a_3)(a_2, a_4)(a_3, a_4)$
$(a_1, a_2)(a_1, a_3)(a_1, a_4)(a_2, a_4)(a_3, a_4)(a_2, \mathrm{BS})(a_1, \underline{a_2})(a_4, \mathrm{BS})(a_1, \mathrm{BS})(a_3, a_1)(a_3, \mathrm{BS})(a_1, a_2)(a_1, a_3)(a_1, a_4)(a_2, a_3)(a_2, a_4)(a_3, a_4)$
$(a_1, a_2)(a_1, a_3)(a_1, a_4)(a_4, \mathrm{BS})(a_2, \underline{a_4})(a_2, \mathrm{BS})(a_1, a_2)(a_4, \mathrm{BS})(a_1, \mathrm{BS})(a_3, a_1)(a_3, \mathrm{BS})(a_1, a_2)(a_1, a_3)(a_1, a_4)(a_2, a_3)(a_2, a_4)(a_3, a_4)$
$(a_1, a_2)(a_1, a_3)(a_4, \underline{\mathrm{BS}})$ <terminé>

Notons que les remarques 2 et 3 ci-dessus permettent d'obtenir une meilleure borne supérieure lorsque les temps de couverture sont différents.

Théorème 3. La complexité au pire de MZP est bornée supérieurement par la somme des temps de couverture des agents de **A***, comptée en événements.

4 Conclusion et perspectives

L'étude du protocole de collecte de données du projet ZebraNet met en évidence un défaut: certaines exécutions ne se terminent pas. La légère modification que nous proposons corrige ce défaut. Néanmoins, la complexité au pire de l'algorithme modifié n'est pas bonne, très éloignée de celle du protocole optimal [BBCK10], qui est de l'ordre de $2\times\mathbf{cv}_{\mathbf{min}}$. Cependant, on peut penser qu'une analyse en moyenne donnerait de meilleurs résultats pour MZP.

D'autre part, il existe d'autres protocoles basés sur l'histoire passée, mais plus élaborés que celui de ZebraNet, comme par exemple PROPHET [LDS04]. Il serait intéressant de les étudier dans le même cadre et de comparer les complexités.

Références

[AADFP04] Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M. J., Peralta, R., Computation in networks of passively mobile finite-state sensors. In PODC, pp. 290-299, 2004.

[AAER07] Angluin, D., Aspnes, J., Eisenstat, D., Ruppert, E., The computational power of population protocols. In DC, 20(4): 279-304, 2007.

[BBCK10] Beauquier, J., Burman, J., Clement J., Kutten, S., On utilizing speed in networks of mobile agents. In PODC, pp. 305-314, 2010.

[BBK09] Bertier M., Busnel Y., Kermarrec A.-M., On Gossip and Population. In SIROCCO, pp. 72-86, 2009.

[BCMRR07] Beauquier, J., Clément, J., Messika, S., Rosaz, L., Rozoy, B., Self-stabilizing counting in mobile sensor networks with a base station. In DISC, pp. 63-76, 2007.

[FJ06] Fischer, M., Jiang, H., Self-stabilizing leader election in networks of finite-state anonymous agents. In OPODIS, LNCS, vol. 4305, pp. 395-409, 2006.

[GR07] Guerraoui, R., Ruppert, E., Even small birds are unique: Population protocols with identifiers. Technical Report CSE-2007-04, York University, 2007

[JOWMPR02] Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L., Rubenstein, D., Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. In ASPLOS, pp. 96-107, 2002.

[LDS04] Lindgren, A., Doria A., Schelén O., Probabilistic Routing in Intermittently Connected Networks. In SAPIR, pp. 239-254, 2004.

[RSHLC08] Rhee, I., Shin, M., Hong, S., Lee, K., Chong, S., On the levy-walk nature of human mobility. In INFOCOM, pp. 924-932, 2008.