



HAL
open science

Comptage et nommage simples et efficaces dans les protocoles de populations symétriques

Joffroy Beauquier, Janna Burman, Simon Clavière

► **To cite this version:**

Joffroy Beauquier, Janna Burman, Simon Clavière. Comptage et nommage simples et efficaces dans les protocoles de populations symétriques. ALGOTEL 2014 – 16èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Jun 2014, Le Bois-Plage-en-Ré, France. pp.1-4. hal-00986109v1

HAL Id: hal-00986109

<https://hal.science/hal-00986109v1>

Submitted on 30 Apr 2014 (v1), last revised 7 May 2014 (v2)

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.

Comptage et nommage simples et efficaces dans les protocoles de populations symétriques

Joffroy Beauquier¹ and Janna Burman¹ and Simon Clavière¹

¹LRI, Université Paris-Sud 11, Orsay, France, {jb,burman,claviere}@lri.fr

Nous améliorons les meilleurs résultats connus, en terme de nombre d'états d'un agent, pour les solutions aux problèmes du *comptage* et du *nommage* dans un modèle voisin des *protocoles de populations*. Pour le comptage, étant donnée une borne P sur le nombre d'agents, le meilleur résultat connu était $\frac{3}{2}P$ états, sous l'hypothèse d'*équité globale*. Nous présentons une solution qui nécessite seulement P états. Pour le nommage, il existe une solution avec $\frac{3}{2}P$ états par agent, sous l'hypothèse d'*équité globale*. Nous présentons une solution à P états, ainsi qu'une solution à $2P$ états sous *équité faible* (les deux solutions sont totalement *auto-stabilisantes*). Enfin, nous décrivons un protocole de *nommage consécutif minimal*.

Keywords: protocoles de populations, comptage, nommage, équité

1 Introduction

Compter et nommer les éléments d'un ensemble sont des tâches fondamentales en informatique, car les applications en sont extrêmement nombreuses. Le domaine qui nous intéresse ici est celui des réseaux de capteurs mobiles, à très grande échelle. Comme les capteurs sont généralement fragiles et que nombre d'entre eux peuvent être victimes d'incidents de fonctionnement, il peut être utile de connaître le nombre de ceux qui sont encore actifs à un moment donné. D'un autre côté, donner des noms différents permet de résoudre des problèmes qui exigent de briser la symétrie initiale dans des réseaux de capteurs anonymes.

Pour traiter ce type de réseaux, on considère ici le modèle des protocoles de populations (noté PP) [AAD⁺06]. Ce modèle considère des agents mobiles à état fini, anonymes, asynchrones, chacun décrit par un système de transitions. Lorsque deux agents se rencontrent, il se produit une transition qui peut modifier leurs états respectifs.

Les problèmes du comptage et du nommage d'agents dans PP ne sont pas nouveaux et plusieurs résultats sont connus. Ces résultats utilisent une borne supérieure P au nombre d'agents et la complexité est exprimée en le nombre d'états nécessaires pour chaque agent, en fonction de P . Pour le problème de nommage, P est une borne inférieure évidente. Elle l'est aussi pour le comptage, comme il l'est prouvé dans [BCM⁺07] sous l'hypothèse d'*équité faible* (pour un algorithme déterministe). Les paramètres du modèle sont d'une part l'hypothèse d'*équité* qui est faite sur les exécutions (*équité faible* ou *globale*), d'autre part la présence ou non d'un agent distingué avec une mémoire non-bornée (la station de base, notée BS), et enfin, la symétrie ou non des règles de transition définissant le protocole. Toute notre étude concerne le cas, plus difficile, de règles symétriques (où, dans une transition, deux agents de même état sont indistinguables).

Sous les hypothèses d'*équité faible*, de présence d'une station de base et de règles symétriques, [BCM⁺07] fournit une solution avec $4P$ états, et une solution à P états dans le modèle asymétrique (que le transformateur de [BCCK08] peut transformer en une solution symétrique en doublant le nombre des états). Dans [KIIW10], le premier résultat de [BCM⁺07] est amélioré à $2P$ (sous *équité faible*), et à $\frac{3}{2}P$ sous *équité globale*. Nous améliorons ici ce dernier résultat en présentant une solution nécessitant seulement P états.

Nous nous intéressons ensuite au problème du nommage (donner des identificateurs différents aux agents), sous les deux hypothèses d'*équité*. Un résultat est connu [KIIW10], qui fournit indirectement un algorithme

de nommage des agents nécessitant $\frac{3}{2}P$ états par agent sous l’hypothèse d’équité globale, avec une station de base devant être initialisée. Sous la même hypothèse d’équité, nous présentons deux solutions auto-stabilisantes (avec initialisation quelconque), l’une est déterministe et l’autre non (mais sans BS), qui ne nécessitent que $P + 1$ et P états par agent, respectivement. Sous équité faible et avec station de base, nous obtenons également une solution auto-stabilisante (pas d’initialisation, ni des agents, ni de BS) avec $2P$ états par agent. Notez que l’algorithme de nommage qui est utilisé comme base de protocole d’élection de leader dans [CIW12] utilise fortement l’asymétrie des règles. Le transformateur de [BCKK08] rendra cet algorithme symétrique, mais en doublant le nombre des états.

Enfin, en combinant nos solutions pour le comptage et le nommage nous proposons un algorithme de nommage consécutif minimal (les identificateurs attribués sont dans $\{1, 2, \dots, n\}$, pour un système avec n agents). Aucun de nos algorithmes ne nécessite d’initialiser les agents (ni même BS, à part le cas de comptage qui est impossible sans initialisation de BS). [BBC14] est la version étendue de cet article.

2 Le modèle

On considère un système constitué de n agents anonymes (n est inconnu des agents). Les variables des agents sont bornées sauf l’une d’entre elles, qui est destinée à recevoir le nom de l’agent (sa taille est déterminée par une borne supérieure au nombre d’agents). Dans certains cas, on introduit un agent distingué à la mémoire non-bornée, appelée *station de base* (BS). Un système est modélisé comme un système de transitions, dont nous adoptons les définitions classiques : *état* d’un agent (la suite des valeurs de ses variables), *configuration* (un vecteur d’états), *transitions* (pas atomique effectué par deux agents se rencontrant et le changement d’états associé), *exécution* (une suite potentiellement infinie de configurations dans laquelle chaque élément est obtenu à partir du précédent par une transition).

Nous notons (x, y) l’interaction (rencontre) entre les agents x et y . Durant l’interaction (x, y) , une *transition symétrique* de la forme $(s_x, s_y) \rightarrow (s'_x, s'_y)$ est exécutée, et si $s_x = s_y$ alors $s'_x = s'_y$. Nous supposons que deux interactions se produisent en séquence.

Pour modéliser la mobilité, il est pratique de considérer les exécutions comme si un ordonnanceur externe choisissait quel couple d’agents participe à la prochaine interaction. Formellement, un *ordonnanceur* est un prédicat O sur les suites d’interactions. Un ordonnanceur O est dit *faiblement équitable* (*weak fairness*) si dans toute séquence d’interactions satisfaisant O , tout couple d’agents (x, y) réalise une infinité d’interactions. O est dit *globalement équitable* (*global fairness*) pour un système S si, étant donnée une exécution e de S ordonnée par O , toute configuration C infiniment souvent accessible (à partir des configurations de e), apparaît infiniment souvent dans e (voir [AAD⁺06] pour les définitions formelles).

Un problème est défini comme une propriété des exécutions. Un protocole de populations A résout un *problème* Pr si et seulement si toute exécution de A satisfait Pr . Le problème du *comptage* est défini par la propriété suivante : la station de base possède une variable $NbAgents$ qui, à partir d’un certain rang, ne varie pas et contient le nombre d’agents du système. Celui du *nommage* est défini par : chaque agent possède une variable ID qui, à partir d’un certain rang, ne varie pas et telle que les valeurs des variables ID de deux agents différents sont différentes.

Mises à part la variable non-bornée, la station de base et l’absence de mécanisme pour briser la symétrie (dans une transition), ce modèle est celui des protocoles de populations originels [AAD⁺06].

3 Comptage

Nous présentons dans cette section un protocole (Alg. 1) permettant de compter les agents (anonymes) d’un réseau, sous l’hypothèse d’équité globale, en utilisant P états différents par agent.

L’idée principale de l’algorithme 1 est la suivante. Quand un agent interagit avec BS et que BS considère que son numéro a le statut “utilisé”, elle attribue à l’agent un nouveau numéro et “libère” l’ancien. Le numéro d’un agent peut donc changer à chaque rencontre avec la station de base. Celle-ci ne peut pas attribuer de nouveau un numéro avant qu’un agent avec le même numéro ne la rencontre. Afin de garantir qu’à partir d’un certain moment tout numéro comptabilisé par la station de base soit réellement le numéro d’un agent (ce pourrait être le cas du fait d’une mauvaise initialisation), la station de base doit être initialisée (mais pas les agents). Dans [BCM⁺07], il est prouvé que cette hypothèse est nécessaire.

L'algorithme de BS utilise un tableau T_{BS} de taille P avec $T_{BS}[\text{numéro}] = 0$ représentant un slot libre, et $T_{BS}[\text{numéro}] = 1$ un slot utilisé. La variable $NbAgents$ de BS contient le nombre d'agents du système, à partir d'un certain rang.

Algorithme 1 Comptage

(sous équité globale, avec P états par agent)

Variabes de BS :

$T_{BS}[0, \dots, P-1]$: tableau de bits, initialisé à 0
 $NbAgents_{BS}$: taille de l'ensemble $\{i : T_{BS}[i] = 1\}$

Variable de l'agent p :

$numéro_p$: entier dans $[0, P-1]$

1: **Quand un agent p interagit avec BS :**
2: **if** ($NbAgents_{BS} \neq P \wedge T_{BS}[numéro_p] = 1$) **then**
3: $T_{BS}[numéro_p] \leftarrow 0$
4: /* attribue un numéro libre */
5: $numéro_p \leftarrow \min\{i \in [0, P-1] : i \neq numéro_p \wedge T_{BS}[i] = 0\}$
6: **end if**
7: $T[numéro_p] \leftarrow 1$

Algorithme 2 Nommage

(auto-stabilisant, sous équité faible, avec $2P$ états par agent)

Variable de BS :

$T_{BS}[1, \dots, 2P-1]$: tableau de bits

Variable de l'agent p :

ID_p : entier dans $[0, 2P-1]$

1: **Quand un agent p interagit avec BS :**
2: **if** ($ID_p = 0$) **then**
3: /* BS estime avoir attribué tout les slots */
4: **if** ($\forall i : T_{BS}[i] = 1$) **then**
5: /* remise à zéro du tableau */
6: $T_{BS} \leftarrow \{0\}$
7: **end if**
8: $ID_p \leftarrow \min\{i : T_{BS}[i] = 0\}$
9: **end if**
10: $T_{BS}[ID_p] \leftarrow 1$
11:
12: **Quand deux agents p et p' interagissent :**
13: **if** $ID_p = ID_{p'} \neq 0$ **then**
14: $ID_p \leftarrow 0, ID_{p'} \leftarrow 0$
15: **end if**

Théorème 1 Sous l'hypothèse d'équité globale, l'algorithme 1 résout le problème du comptage avec P états par agent.

Schéma de la preuve. Notons que grâce à l'initialisation de BS, si $T_{BS}[i] = 1$, il existe au moins un agent x tel que $numéro_x = i$, ce qui implique $NbAgents_{BS} \leq n$. Considérons une configuration C' dans laquelle $NbAgents_{BS} \leq n$. On s'intéresse tout d'abord à l'ensemble des $numéro$ utilisés par au moins un agent. Pour chaque valeur i de ces $numéro$, on choisit de faire interagir avec BS un seul agent x tel que $numéro_x = i$ et $T_{BS}[i] = 0$. Une fois ces interactions effectuées, $T_{BS}[i] = 1$ pour chaque slot i , tel qu'il existe au moins un agent avec $numéro = i$. Les slots i sont utilisés et BS ne peut pas les attribuer à un autre agent. Ces slots ne peuvent être "libérés" que si un agent avec $numéro = i$ interagit avec BS. Maintenant, pour chaque slot i utilisé par plus d'un agent, on considère l'ensemble des agents S_i tel que pour chaque agent $x \in S_i$, $numéro_x = i$. Pour chaque ensemble, on choisit de faire interagir chaque agent de l'ensemble, une fois avec BS. Une fois ces interactions effectuées, chacun de ces agents a un $numéro$ unique, tel que $T_{BS}[numéro] = 1$. Le nombre de slots utilisés est alors égal aux nombre d'agents avec un $numéro$ unique. Comme tout les agents ont un $numéro$ unique, on atteint une configuration C dans laquelle $NbAgents_{BS} = n$. Par conséquent, si une configuration C' est accessible infiniment souvent, du fait de l'hypothèse d'équité globale, une configuration C dans laquelle $NbAgents_{BS} = n$ l'est aussi. De plus, à partir d'une telle configuration, aucune interaction ne peut modifier la valeur de $NbAgents_{BS}$. Ceci entraîne que les seules configurations accessibles infiniment souvent dans une exécution sont telles que $NbAgents_{BS} = n$. \square

4 Nommage

Nommage auto-stabilisant, sous équité faible, avec BS et $2P$ états - Alg. 2. Cet algorithme attribue un identifiant $ID \in [1, 2P-1]$, à chaque agent *anonyme* ($ID = 0$) interagissant avec la station de base. Chaque ID est associé à un *slot* dans un tableau T_{BS} de la station de base ; $T_{BS}[x] = 0$ signifie que BS

considère l' ID x comme libre, $T_{BS}[x] = 1$ signifie que BS estime que cet ID est utilisé par un agent. Les ID libres sont attribués par ordre croissant de disponibilité.

Quand un agent anonyme interagit avec la station de base, le plus petit ID libre lui est attribué (cf. ligne 8). Le slot correspondant est mis à jour, $T[x] \leftarrow 1$ (cf. ligne 10). Comme l'algorithme ne requiert aucune initialisation, il peut exister parmi la population plusieurs agents avec un même ID , et il peut également exister des slots indiqués comme attribués par la station de base, mais dont aucun agent ne porte l' ID . Pour "corriger" le premier type d'erreur, lorsque deux agents avec le même ID se rencontrent, l'algorithme remet ces deux ID à 0. Pour "corriger" le second type d'erreur, la station de base effectue une remise à zéro de son tableau T_{BS} lorsqu'un agent anonyme se présente et qu'aucun slot n'est disponible. Une remise à zéro libère l'ensemble des slots (cf. ligne 6).

Théorème 2 *Sous équité faible, l'algorithme 2 résout le problème du nommage auto-stabilisant.*

Schéma de la preuve. S'il existe des agents avec un même ID , ils finiront par se rencontrer, du fait de l'équité faible, et deviendront des agents anonymes (lignes 13 – 15). Tant qu'il existe des agents anonymes, BS leur attribue le premier slot libre (ligne 8). BS ne peut ainsi attribuer un slot déjà attribué qu'après avoir attribué les $2P - 2$ autres slots. Notez que il y a au plus $n - 1$ agents avec des ID différents dans une configuration où le nommage n'est pas réalisé. Donc, entre deux attributions d'un même slot, BS ne peut pas attribuer plus de $n - 2$ ID déjà existants ; et, s'ils sont attribués, chacun de n slots restants n'est utilisé que par un et un seul agent. BS attribue donc un ID distinct à chaque agent entre deux attributions d'un même slot. Quand tous les agents ont des ID distincts, la ligne 10 est la seule à être exécutée, et donc l' ID de chaque agent ne change plus. \square

Nommage auto-stabilisant, sous équité globale, avec BS et $P + 1$ états. Sous l'hypothèse d'équité globale, l'algorithme 2 est également correct en fixant la taille du tableau T_{BS} à P avec seulement $P + 1$ états par agent. Voir [BBC14] pour la preuve.

Nommage optimal (avec P états) auto-stabilisant, non-déterministe, sous équité globale, sans BS. L'algorithme est très simple. Chaque agent a une seule variable $ID \in [0, P - 1]$. À chaque interaction entre deux agents avec le même ID , chacun d'entre eux choisit de manière non-déterministe un nouvel ID . Dans cet algorithme, à partir de toute configuration, une configuration où tous les agents ont des ID uniques est toujours accessible (on fait interagir les agents avec le même ID). L'hypothèse d'équité globale assure qu'une telle configuration est atteinte, à partir de laquelle les ID ne peuvent plus changer.

Nommage minimal. Le nommage minimal (consécutif) est un cas particulier du nommage dans lequel, à partir d'un certain point, les identificateurs sont dans l'ensemble $\{1, 2, \dots, n\}$. Pour obtenir un tel nommage, il est possible de composer l'algorithme de comptage (algorithme 1) avec l'algorithme de nommage auto-stabilisant avec P états, ou un autre avec $P + 1$ états (donnés dans les paragraphes précédents), en utilisant la technique de composition équitable [DIM93].

Références

- [AAD⁺06] D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed computing*, 18(4) :235–253, 2006.
- [BBC14] J. Beauquier, J. Burman, and S. Clavière. Comptage et nommage simples et efficaces dans les protocoles de populations symétriques. Technical report, INRIA, 2014. <http://hal.inria.fr/hal-00948186>.
- [BCKK08] O. Bournez, J. Chalopin, J. Cohen, and X. Koegler. Playing with population protocols. In *CSP*, pages 3–15, 2008.
- [BCM⁺07] J. Beauquier, J. Clement, S. Messika, L. Rosaz, and B. Rozoy. Self-stabilizing counting in mobile sensor networks with a base station. In *Distributed Computing*, pages 63–76. 2007.
- [CIW12] S. Cai, T. Izumi, and K. Wada. How to prove impossibility under global fairness : On space complexity of self-stabilizing leader election on a population protocol model. *Theory Comput. Syst.*, 50(3) :433–445, 2012.
- [DIM93] S. Dolev, A. Israeli, and S. Moran. Self-stabilization of dynamic systems assuming only read/write atomicity. *Distributed Computing*, 7(1) :3–16, 1993.
- [KIW10] K. Kinpara, T. Izumi, T. Izumi, and K. Wada. Improving space complexity of self-stabilizing counting on mobile sensor networks. In *OPODIS*, pages 504–515. 2010.