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

Joffroy Beauquier, Janna Burman, Simon Claviére

▶ **To cite this version:**

Joffroy Beauquier, Janna Burman, Simon Claviére. Comptage et nommage simples et efficaces dans les protocoles de populations symétriques. [Rapport de recherche] 2014. hal-00948186v1

HAL Id: hal-00948186

https://inria.hal.science/hal-00948186v1

Submitted on 17 Feb 2014 (v1), last revised 7 May 2014 (v3)

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

J. Beauquier[1] and J. Burman[1] and S. Clavière[1]

[1]*LRI, 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 $\frac{3}{2}P$, sous l'hypothèse d'*équité globale*. Nous présentons une solution optimale qui nécessite seulement P états. Pour le nommage, il exist une solution avec $\frac{3}{2}P$ états par agent, sous 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 *comptage approximatif* nécessitant $O(\log(P))$ états et un protocole de *nommage consécutif minimal*.

Keywords: Protocoles de populations, comptage, nommage, équité

1 Introduction

Compter et nommer les éléments d'un ensemble, de manière exacte ou approximative, est une tâche fondamentale 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, même de façon approximative, 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 de comptage et de 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$^+$07]. De plus, le comptage est impossible dans les cas où ni BS, ni les agents ne sont initialisés [BCM$^+$07]. Les paramètres du modèle sont d'une part l'hypothèse d'équité qui est faite sur les exécutions (équité faible ou équité 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 définissant le protocole. Toute notre étude concerne le cas, plus difficile, de règles symétriques.

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 est amélioré à $2P$ (sous équité faible), et à $3/2P$ sous équité globale. Nous améliorons ici ce dernier résultat en présentant une solution nécessitant seulement P états par agent. Cette solution est optimale. Puis, dans un modèle sans station de base, nous présentons un protocole de comptage approximatif, dont nous savons borner l'erreur, avec seulement $\log(P)$ états par agent.

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. Nous présentons une solution avec initialisation quelconque, sans station de base, qui ne nécessite que P états par agent dans les mêmes conditions. Sous l'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.

Enfin, en combinant nos solutions pour le comptage et le nommage nous proposons un algorithme de nommage consecutive minimal (les identificateurs attribués sont dans $\{1,2,\ldots,n\}$, pour un système avec n agents).

Aucun de nos algorithmes ne nécessite d'initialiser les agents (ni même BS dans certains cas).

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 $(s_x,s_y) \to (s'_x,s'_y)$ est exécutée. 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, dans toute exécution e de S ordonnancée par O, si une configuration C peut être obtenue (accessible) infiniment souvent dans e, alors C apparaît infiniment souvent dans e (voir [AAD$^+$06] pour les définitions formelles).

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, l'absence de mécanisme pour briser la symétrie et la station de base, ce modèle est celui des protocoles de populations originels [AAD$^+$06].

Dans la suite, classiquement, nous ne présentons pas les protocoles sous la forme de systèmes de transitions, mais sous celle, équivalente, de pseudo-code.

3 Comptage

Nous présentons dans cette section un protocole permettant de compter les agents (anonymes) d'un réseau, en utilisant P états différents par agent (ce qui est optimal). L'algorithme est correct sous les deux hypothèses d'équité (faible et globale). Contrairement aux algorithmes de comptage déjà connus dans le modèle PP, l'algorithme proposé ne réalise pas déjà un nommage, pour compter les ID. Au contraire, chaque fois qu'un agent interagit avec la station de base, il acquiert un nouveau numéro. L'absence de passage par un nommage nous permet d'abaisser la complexité en mémoire des agents. L'idée principale de l'algorithme 1 réside dans l'attribution par la station de base, d'un nouveau *numéro* $\in [0,P-1]$ à chaque agent qui la rencontre. Le numéro d'un agent change donc à chaque rencontre avec la station de base. Celle-ci ne peut

pas ré-affecter 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). On peut prouver que cette hypothèse est nécessaire.

L'algorithme de BS utilise un tableau T_{BS} de taille P avec $T_{BS}[numéro] = 0$ représentant un slot libre, et $T_{BS}[numéro] = 1$ un slot utilisé.

Algorithme 1 Comptage
(sous équité globale, avec P états par agent)

Variables de BS :
$T_{BS}[0,\dots P-1]$: tableau de bits, initialisé à 0
$NbAgent_{BS}$: 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** $(NbAgent_{BS} \neq P \wedge T_{BS}[numéro_p] = 1)$ **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: **Lorsque deux agents p et p' interagissent :**
13: **if** $ID_p = ID_{p'}$ **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.*

Schéma de la preuve. Notons que grâce à l'initialisation de BS, si $T_{BS}[i] = 1$, il existe au moins un agent tel que $numéro_x = i$, ce qui implique $NbAgents \leq n$. On montre qu'à partir d'une configuration C' accessible infiniment souvent dans une exécution de l'algorithme 1, il y a une configuration accessible C dans laquelle $NbAgents_{BS} = n$

Considérons donc une configuration C' dans laquelle $NbAgents \leq n$. 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' et 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 configu-

ration, 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 $NbAgents = n$.

Ainsi, sous l'hypothèse d'équité globale, l'algorithme 1 résout le problème du comptage en utilisant P états par agent. □

3.1 Comptage approximatif logarithmique

Dans les situations où la mémoire disponible en chaque agent est une ressource vraiment critique, on peut envisager une modification de la spécification du problème du comptage, en remplaçant comptage exact par comptage approximatif. Il y a certainement de nombreuses situations dans lesquelles avoir un ordre de grandeur du nombre d'agents est amplement suffisant. Une solution approximative est encore plus satisfaisante si on peut borner l'erreur par rapport à la solution exacte.

C'est une telle solution dont nous décrivons les grandes lignes dans cette section. Cette solution utilise des propriétés et des techniques connues, mais qui, à notre connaissance, n'ont jamais été combinées. Tout d'abord, elle fait appel à la caractérisation des protocoles de populations en terme de calculabilité et d'arithmétique de Pressburger (cf. [AAD$^+$06]). Ensuite elle utilise une technique voisine dans son esprit du comptage approximatif de Morris [Mor78], qui reçu une preuve et une étude de complexité analytique de la part de Flajolet [FNM85]. Cette étude permet de borner l'erreur par rapport au comptage exact. La mise en œuvre de la méthode de comptage approximatif ne nécessite pas de station de base, demande seulement une équité faible et une hypothèse supplémentaire faible sur les agents (modèle des homonymes) : chaque agent a reçu préalablement une couleur, bleu ou rouge, stockée comme constante et il y a dans le système au moins deux agents bleus et deux agents rouges. Hormis cette contrainte, les nombres d'agents bleus et d'agents rouges sont quelconques, et bien entendu inconnus des agents.

Nous combinons plusieurs protocoles. Nous utilisons d'abord le fait que le ratio (arrondi à l'entier immédiatement inférieur) entre nombre d'agents bleus et nombre d'agents rouges est Pressburger et est donc calculable par un protocole de population (au sens de [AAD$^+$06]). On supposera que chaque agent dispose d'une variable R, dont la valeur se stabilise au ratio ci-dessus. Pour la clarté de la suite on choisira $R = 1$ (autant d'agents bleus et rouges). Chaque agent exécute un deuxième protocole, qui simule un tirage aléatoire (ici équiprobable parce que $R = 1$), en supposant que les agents ont un mouvement aléatoire. Après avoir, à un instant quelconque, rencontré un agent ayant la même couleur que lui, cet agent décide de compter le nombre d'interactions avec d'autres agents, jusqu'à rencontrer de nouveau un agent ayant la même couleur que lui et range le résultat dans une variable C. Ce comptage local est réalisé une seule fois. Le résultat de Morris nous dit que la valeur maximum de C est une bonne approximation du logarithme du nombre d'agents. Une formule bornant l'erreur, pour le cas particulier de deux couleurs, mais aussi pour des cas plus généraux, est donnée dans [FM83]. Il suffit donc que chaque agent exécute un troisième protocole, très simple, pour obtenir ce maximum. La composition parallèle des trois protocoles fournit un protocole qui se stabilise vers une valeur approchée du logarithme du nombre d'agents et qui n'utilise que $O(\log(P))$ états par agent.

Il n'est pas difficile d'imaginer comment cette méthode se généralise à plusieurs couleurs et des nombres d'agents différents pour chaque couleur. Dans la pratique, la couleur peut être obtenue comme dernier chiffre d'un numéro de série ou de la date de fabrication. D'autre part, le caractère aléatoire de la mobilité des agents a été observé expérimentalement pour de nombreuses populations humaines ou animales (étudiants d'un campus [Col07], participants à une conférence [HCS$^+$05], populations d'albatros, de bourdons et de daims [EPW$^+$07]).

4 Nommage

Nous présentons dans cette section un algorithme auto-stabilisant (sans l'initialisation de BS ni des agents) permettant le nommage de n agents anonymes, utilisant $2P$ états différents. L'algorithme est correct sous l'hypothèse d'équité faible. L'idée principale de ce protocole, mis en œuvre par l'algorithme 2 est d'affecter un identifiant $\in [1, 2P-1]$, 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 peut ainsi attribuer un slot déjà attribué uniquement après avoir attribué les $2P-2$ autres slots. Entre deux attributions d'un même slot, comme il y a au plus $(n-1)$ agents avec des *ID* différents dans un configuration où le nommage n'est pas réalisé, 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. □

Nommage avec $P+1$ 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.

Considérons une configuration C dans laquelle les agents n'ont pas d'indificateurs distincts et montrons que de C on peut atteindre une configuration où ils en ont.

D'abord, il est facile a voir que dans une exécution de l'algorithme 2 (et dans sa version adaptée), si aucune remise à zéro du tableau de BS ne se produit, à partir d'un certain point, l'algorithme finit par résoudre le problème du nommage (tous les agents ont des indificateurs distincts). Supposons maintenant une remise à zéro, et soit C la configuration juste après. À partir de C, si on choisit de faire interagir d'abord tous les agents avec $ID \neq 0$ avec BS, et si on choisit ensuite toutes les autres interactions, on finira par nommer tous les agents de manière distincte (et *neq*0). Par conséquent, une telle configuration où tous les agents ont des indificateurs distincts (et *neq*0) est toujours accessible. Par équité globale, une telle configuration est atteinte, et à partir de là, les identificateurs ne changent plus (ligne 10). Donc, l'algorithme adapté résout le problème du nommage, en utilisant P+1 états par agent.

Nommage sans BS avec P états. Sous l'hypothèse d'équité globale, on propose ici un algorithme très simple, optimal pour le nombre d'états par agent (P états), sans BS et auto-stabilisant. L'algorithme proposé est non-déterministe, mais son comportement étant "élastique" au sens de [AAFJ08], il peut être convertit en un algorithme déterministe en utilisant le transformateur de [AAFJ08].

Dans l'algorithme proposé, 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 que une telle configurations est atteinte et à partir desquelle les ID ne peuvent plus changer.

Nommage (consécutif) minimal. Le nommage minimal est un cas particulier du nommage dans lequel, à partir d'un certain point, les identificateurs sont dans l'ensemble $\{1, 2, ..., n\}$. Pour obtenir un tel nommage, il est possible de composer l'algorithme de comptage (algorithme 1) avec l'algorithme de nommage auto-stabilisant optimal avec P états, donné dans le paragraphe précédent.

On utilise la technique de composition équitable des protocoles [DIM93]. Suivant cette technique, l'algorithme optimal de nommage s'exécute en parallèle avec l'algorithme de comptage, et utilise la sortie

NbAgents pour définir la borne P. À partir du moment où *NbAgents* se stabilise à n, l'algorithme de nommage va se stabiliser avec $P = n$ et nommer tous les agents avec des ID uniquement dans $[1, n]$.

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.

[AAFJ08] D. Angluin, J. Aspnes, M. J Fischer, and H. Jiang. Self-stabilizing population protocols. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 3(4) :13, 2008.

[BCCK08] 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. Springer, 2007.

[Col07] The Dartmouth wireless trace archive - http ://crawdad.cs.dartmouth.edu/. Dartmouth College, 2007.

[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.

[EPW+07] Andrew M. Edwards, Richard A. Phillips, Nicholas W. Watkins, Mervyn P. Freeman, Eugene J. Murphy, Vsevolod Afanasyev, Sergey V. Buldyrev, M. G. E. da Luz, E. P. Raposo, H. Eugene Stanley, and Gandhimohan M. Viswanathan. Revisiting levy flight search patterns of wandering albatrosses, bumblebees and deer. *Nature*, 449 :1044–1048, October 2007.

[FM83] P. Flajolet and G. N. Martin. Probabilistic counting. In *FOCS*, pages 76–82, 1983.

[FNM85] P. Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *Journal of computer and system sciences*, 31(2) :182–209, 1985.

[HCS+05] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket switched networks and human mobility in conference environments. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, WDTN '05, pages 244–251, New York, NY, USA, 2005. ACM.

[KIIW10] 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.

[Mor78] R. Morris. Counting large numbers of events in small registers. *Commun. ACM*, 21(10) :840–842, 1978.