# *Rassemblement byzantin dans les réseaux* †

Sébastien Bouchard[1] et Yoann Dieudonné[2] et Bertrand Ducourthial[3]

[1] *Sorbonne Université, CNRS, INRIA, Laboratoire d'Informatique de Paris 6, LIP6, DELYS, F-75005 Paris, France*
[2] *Laboratoire Modélisation Information et Systèmes (MIS), Université de Picardie Jules Verne, France*
[3] *Heudiasyc, CNRS et Université de Technologie de Compiègne, Compiègne, France*

Au moins deux agents mobiles se déplacent au sein d'un réseau, de sommet en sommet en traversant ses arêtes et doivent accomplir la tâche du rassemblement qui consiste à les réunir en un même sommet. Un adversaire choisit les sommets initiaux des agents et assigne à chacun un entier positif (appelé identifiant). Au début de toute exécution, chaque agent connaît son identifiant mais ne connaît ni les identifiants des autres agents, ni leurs positions. Les agents se déplacent en rondes synchrones et ne peuvent communiquer entre eux que quand ils se trouvent sur un même sommet. Parmi les agents, il y a au plus $f$ agents byzantins (les autres agents sont qualifiés de bons). Un agent byzantin peut choisir n'importe quel port pour se déplacer et transmettre n'importe quelle information aux autres agents, en particulier en mentant à propos de son identifiant, en adoptant celui d'un autre agent ou en en choisissant un tout nouveau.

*Quel est le nombre minimum $\mathcal{M}$ de bons agents qui garantit le rassemblement déterministe de chacun d'entre eux avec détection de terminaison ?* Nous apportons des réponses exactes à cette question d'une part quand les agents connaissent initialement la taille du réseau et d'autre part quand ce n'est pas le cas. Dans le premier cas, nous prouvons que $\mathcal{M} = f+1$ et dans le second, nous montrons que $\mathcal{M} = f+2$.

**Mots-clefs :** rendez-vous, algorithme déterministe, agent mobile, faute byzantine

## 1 Introduction

### 1.1 Modèle et problème

Le système distribué considéré dans cet article est un groupe d'agents mobiles initialement placés par un adversaire sur des sommets quelconques (pas nécessairement tous distincts) d'un réseau modélisé par un graphe simple, fini, bidirectionnel et connexe $G = (V, E)$. Nous notons $n$ le nombre de sommets de ce graphe, et l'appelons aussi taille du graphe ou taille du réseau. Les sommets sont anonymes, ce qui signifie qu'ils n'ont rien s'apparentant à un identifiant permettant de les distinguer les uns des autres. Par contre, les arêtes incidentes à un même sommet peuvent être distinguées les unes des autres grâce à un numéro de port compris entre 0 et le degré du sommet $-1$. Aucune cohérence n'est supposée entre les deux numéros de port d'une même arête.

Les agents sont initialement dormants. Ils doivent être réveillés pour interagir avec les autres agents et se déplacer. Ce réveil peut avoir lieu de deux façons. Premièrement, l'adversaire réveille certains agents (au moins un), à des moments éventuellement différents. Deuxièmement, un agent se réveille dès qu'un agent réveillé se trouve sur le même sommet que lui.

Le temps est discrétisé en une suite infinie de rondes. À chaque ronde, chaque agent réveillé exécute un algorithme déterministe pour choisir s'il reste sur place, ou traverse l'une des arêtes incidentes à son sommet actuel. L'algorithme exécuté est le même pour tous les agents. Seule son entrée, dont la nature est détaillée ci-après, dépend des agents.

En plus de leurs sommets initiaux, l'adversaire assigne initialement à chaque agent un identifiant unique. Chaque agent connaît son identifiant mais n'a aucune connaissance à priori de ceux des autres agents. Quand il se trouve sur un sommet, chaque agent connaît le degré de ce dernier, ainsi que (le cas échéant) le numéro du port par lequel il est entré.

† Cet article est un résumé étendu de [BDD16]

Quand plusieurs agents se trouvent dans un même sommet $v$ à une ronde $r$, ils voient pour chaque agent $a$ dans $v$ à $r$, son identifiant et les informations que $a$ choisit de partager à cette ronde. Les informations partagées par $a$ avec chacun des autres agents présents sur $v$ à cette ronde sont les mêmes. Néanmoins, quand deux agents ne partagent pas le même sommet, ils ne peuvent ni se voir, ni communiquer.

Toujours à propos des informations dont disposent les agents, nous étudions deux scénarios, un premier dans lequel chaque agent connaît initialement la taille du graphe, et un second dans lequel il n'a même pas une borne supérieure sur celle-ci.

Par ailleurs, nous supposons que parmi les agents, jusqu'à $f$ sont byzantins (la valeur de $f$ est connue des agents). Un agent byzantin a une grande capacité de nuisance : il peut choisir n'importe quel port pour se déplacer et transmettre n'importe quelle information aux autres agents, en particulier en mentant à propos de son identifiant, en adoptant celui d'un autre agent ou en en choisissant un tout nouveau. Le cas de la faute byzantine est particulièrement intéressant puisqu'il s'agit de la pire faute que puisse subir un agent. Tous les agents qui ne sont pas byzantins sont qualifiés de bons.

Ainsi, à chaque ronde, l'entrée de l'algorithme exécuté par un bon agent $a$ est constituée des informations initialement à disposition de l'agent (son identifiant, la valeur de $f$ et éventuellement, en fonction du scénario la taille du réseau), et de l'historique à jour de tout ce que $a$ a vu et appris depuis son réveil.

Nous considérons la tâche du rassemblement $f$-Byzantin définie comme suit. Dans un réseau où au plus $f$ agents sont byzantins, l'adversaire réveille au moins un bon agent, et tous les bons agents doivent, à terme, se trouver sur le même sommet à la même ronde, déclarer simultanément la tâche accomplie et s'arrêter. Il convient de mentionner qu'on ne peut pas attendre des byzantins qu'ils coopèrent puisqu'ils peuvent par exemple toujours refuser de se trouver avec d'autres agents. Ainsi, rassembler tous les bons agents avec détection de terminaison est l'exigence la plus forte qu'on puisse émettre dans ce contexte.

Quel est le nombre minimum $\mathcal{M}$ de bons agents qui garantit le rassemblement $f$-byzantin ? De prime abord, cette question peut sembler sans intérêt, puisqu'après tout, les bons agents sont peut-être capables de se rassembler quelque soit le nombre de byzantins. Cependant, ce n'est pas le cas, comme l'étude qui a introduit cette question l'a mis en évidence dans [DPP14].

## 1.2 État de l'art

La littérature à propos du problème du rassemblement est conséquente. Cela est dû au grand nombre de variations possibles à propos des hypothèses sous lesquelles on peut l'étudier.

Cependant, avant nos propres travaux, le rassemblement dans des graphes quelconques en présence d'agents byzantins n'avait été considéré que dans [DPP14]. En plus d'introduire la question qui nous intéresse dans cet article, les auteurs de cet article l'étudient eux-mêmes dans quatre scénarios différents, correspondant à différentes variations à propos des hypothèses faites sur les capacités des agents. Dans deux scénarios, ils considèrent que les agents byzantins sont incapables de mentir à propos de leur identifiant et dans les autres, comme nous le faisons dans cet article, qu'ils en sont capables. Dans chacun de ces deux cas, les auteurs de [DPP14] s'intéressent à l'impact de la connaissance de la taille du réseau en étudiant deux sous-cas, le premier dans lequel cette connaissance est accordée aux bons agents, et le second dans lequel elle ne l'est pas.

Quand les byzantins sont incapables de mentir sur leur identifiant, ils montrent que le nombre minimum $\mathcal{M}$ de bons agents qui garantit le rassemblement $f$-byzantin est précisément 1 dans les réseaux de taille connue, et $f+2$ dans ceux de taille inconnue. La preuve que ces nombres de bons agents sont suffisants repose sur des algorithmes utilisant un mécanisme de "blacklist", c'est-à-dire de liste d'identifiants d'agents ayant eu un comportement "incohérent". Bien sûr, ce mécanisme ne peut plus être utilisé quand les agents byzantins peuvent mentir sur leur identifiant et en particulier voler celui d'un autre agent.

Quand les byzantins sont capables de mentir à propos de leur identifiant, les auteurs de [DPP14] ne donnent pas la valeur exacte de $\mathcal{M}$ mais seulement un encadrement de celle-ci. Plus précisément, quand la taille du réseau est connue des agents, ils présentent un algorithme déterministe pour le rassemblement $f$-byzantin nécessitant qu'il y ait au moins $2f+1$ bons agents dans le réseau, et donnent une borne inférieure de $f+1$ sur $\mathcal{M}$ en montrant que si le nombre de bons agents n'est pas plus grand que $f$, il existe des graphes dans lesquels les bons agents ne peuvent pas se rassembler avec détection de terminaison. Quand la taille du réseau est inconnue, ils procèdent de façon similaire mais montrent des bornes différentes. Ils donnent

un algorithme fonctionnant avec une équipe d'au moins $4f+2$ bons agents, et montrent que $f+2$ est une borne inférieure sur $\mathcal{M}$. Cependant, la question de la valeur exacte de $\mathcal{M}$ dans chacun de ces deux derniers scénarios est laissée ouverte.

### *1.3 Nos résultats*

Dans cet article qui est une synthèse des travaux que nous avons présentés dans [BDD16], nous répondons à cette question ouverte en prouvant que les bornes inférieures de $f+1$ et $f+2$ prouvées dans [DPP14] sont aussi des bornes supérieures. Plus précisément, nous concevons des algorithmes déterministes permettant de rassembler tous les agents si ceux-ci sont au moins $f+1$ quand la taille du réseau est initialement connue des agents, et au moins $f+2$ quand celle-ci leur est initialement inconnue.

Mettre nos résultats en perspective avec les scénarios considérés dans [DPP14] dans lesquels les agents byzantins ne peuvent pas mentir sur leur identifiant révèle une propriété intéressante. Dans cette variante, $\mathcal{M}=1$ dans les réseaux de taille connue et $\mathcal{M}=f+2$ dans les réseaux de taille inconnue. Ainsi, quand la taille du réseau est connue, la capacité des agents byzantins à mentir sur leur identifiant impacte significativement la valeur de $\mathcal{M}$. Cependant, dans le cas général où la taille est inconnue, ce n'est plus le cas puisque $\mathcal{M}=f+2$ que les byzantins puissent ou non mentir à propos de leur identifiant.

| **Référence** | **Agents byzantins ne pouvant pas changer d'identifiant** | | **Agents byzantins pouvant changer leur identifiant** | |
|---|---|---|---|---|
| | **Taille du graphe connue** | **Taille du graphe inconnue** | **Taille du graphe connue** | **Taille du graphe inconnue** |
| [DPP14] | Borne supérieure et inférieure : 1 | Borne supérieure et inférieure : $f+2$ | Borne supérieure : $2f+1$<br>Borne inférieure : $f+1$ | Borne supérieure : $4f+2$<br>Borne inférieure : $f+2$ |
| Cet article | | | Borne supérieure : $f+1$ | Borne supérieure : $f+2$ |

**TABLE 1:** Anciennes et nouvelles bornes sur le nombre minimum de bons agents garantissant le rassemblement $f$-byzantin.

## 2 Intuition de l'algorithme quand la taille du graphe est connue

Par manque de place, nous ne présentons que l'intuition de l'un de nos deux algorithmes, celui pour le cas où la taille du réseau est connue. Néanmoins, l'idée du second algorithme est proche de celle du premier, les différences relèvent essentiellement de détails techniques dans le but de pallier à l'ignorance initiale de la taille du graphe.

Imaginons que nous nous trouvons dans la situation idéale suivante. Tous les agents sont réveillés par l'adversaire au même moment. Mais aussi, chaque agent reçoit en entrée, en plus de son identifiant et de la taille du réseau $n$, un paramètre $\rho=(G^*,L^*)$ correspondant à la configuration initiale des agents dans le graphe tel que :

- $G^*$ est une représentation du graphe $G$ avec tous les numéros de port dans laquelle chaque sommet a un identifiant distinct dans l'ensemble $\{1;\ldots;n\}$ (ces identifiants n'apparaissent pas dans $G$).
- $L^*=\{(v_1,l_1),(v_2,l_2),\ldots,(v_k,l_k)\}$ ($k\geq f+1$) où $(v_i,l_i)\in L^*$ si et seulement si un bon agent ayant $l_i$ pour identifiant se trouve initialement sur le sommet de $G$ ayant pour identifiant $v_i$ dans $G^*$.

Dans une situation aussi idéale, le rassemblement de tous les bons agents peut facilement être accompli en s'assurant que chaque bon agent se rende sur le sommet $v$ où l'agent avec le plus petit identifiant se trouve initialement. Chaque agent peut effectuer ce déplacement grâce aux informations contenues dans $\rho=(G^*,L^*)$ et à son identifiant. Bien sûr, tous les bons agents n'atteignent pas forcément $v$ au même moment. Cependant, en utilisant à nouveau $\rho$ et grâce au fait que tous les agents sont réveillés à la même ronde, chaque bon agent peut calculer le temps restant à attendre avant d'être sûr que tous les bons agents sont sur le sommet $v$.

Malheureusement, les agents ne sont pas dans une situation aussi idéale. Tout d'abord, tous les bons agents ne sont pas nécessairement réveillés par l'adversaire, et pour ceux qui le sont, cela n'a pas forcément lieu à la même ronde. Ensuite, les bons agents ne reçoivent pas $\rho$ en entrée de l'algorithme.

Nous surmontons la première difficulté en utilisant une traversée du graphe comme première instruction de notre algorithme. Par traversée du graphe, nous désignons l'utilisation de la procédure que nous appelons

*EXPLO* qui est un corollaire de [Rei05]. Étant donné un entier positif $N$ quelconque, cette procédure permet à un agent de visiter tous les sommets de n'importe quel graphe d'au plus $N$ sommets, depuis n'importe quel sommet de ce dernier, en $T(EXPLO(N))$ rondes. Ainsi, si on lui fournit la taille $n$ du graphe, cette procédure permet de réveiller tous les agents rencontrés encore dormants, tout en assurant que le délai entre les réveils de deux agents quelconques est au plus $T(EXPLO(n))$. Les agents sont "presque synchronisés" et les périodes d'attente peuvent être ajustées en conséquence.

Il est plus compliqué de se passer de la connaissance de $\rho$. Pour cela, nous utilisons une stratégie inspirée de l'idée introduite dans [CLP12] pour le cas de deux agents évoluant dans un environnement sans fautes. Les agents font des hypothèses qui sont "testées" une à une. Plus précisément, soit $\mathcal{P}$ l'ensemble de toutes les configurations $\rho_i = (G_i^*, L_i^*)$ telles que $G_i^*$ est un graphe connexe à $n$ sommets et $|L_i^*| \geq f+1$. Soit $\Theta = (\rho_1, \rho_2, \rho_3, \ldots)$ une énumération de $\mathcal{P}$ sur laquelle tous les bons agents sont d'accord. Chaque agent procède en phases numérotées $1, 2, 3, \ldots$. Durant la phase $i$, chaque agent suppose que $\rho = \rho_i$ et agit comme il l'aurait fait dans la situation idéale.

Ainsi, il essaie d'aller sur le sommet qu'il suppose correspondre au sommet $v$ (où l'agent de plus petit identifiant se trouve initialement). Pour diverses raisons, quand $\rho_i \neq \rho$, certains agents peuvent être incapables d'effectuer ce déplacement. Par conséquent, ces agents vont considérer, à raison, que $\rho_i$ est différent de $\rho$. Néanmoins, que $\rho_i$ soit égal à $\rho$ ou non, certains bons agents sont susceptibles d'atteindre un sommet tel qu'ils n'ont aucune raison de penser qu'il ne s'agit pas de $v$ (et donc aucune raison de penser que $\rho_i \neq \rho$). Il serait dangereux qu'en arrivant sur le sommet qu'ils croient être $v$, ces bons agents voient chacun des $|L_i^*|$ identifiants de $\rho_i$ (éventuellement, avec "l'aide" des byzantins). En effet, il serait tentant de considérer à ce moment-là que le rassemblement est accompli. Cependant, cela peut ne pas être le cas si $\rho_i \neq \rho$.

Pour parer à ce problème, l'idée est d'envoyer les bons agents pensant que $\rho_i = \rho$ récupérer tous les (éventuels) autres bons agents pour qui $\rho_i \neq \rho$ grâce à une traversée du graphe. Pour cela, un agent convaincu que $\rho_i \neq \rho$ attendra un certain nombre de rondes dans le but de laisser aux bons agents venant le récupérer (s'ils existent) suffisamment de temps pour cela. Pour que cette récupération puisse avoir lieu, il est aussi important que les bons agents en attente ne soient pas récupérés par n'importe quel groupe, et en particulier pas par ceux ne comprenant que des agents byzantins. Ainsi, notre algorithme est conçu de telle sorte qu'à chaque phase, au plus un groupe, appelé *tour* et constitué d'au moins $f+1$ agents, soit reconnaissable sans ambiguïté par tous, et autorisé à récupérer les autres agents via une traversée du graphe.

Quand une tour termine l'exécution de la procédure $EXPLO(n)$ durant une phase $i$, notre algorithme garantit que tous les bons agents sont réunis et déclarent que le rassemblement est effectué au même moment (que la configuration supposée $\rho_i$ soit la bonne ou pas). De plus, à chaque phase, si aucune tour n'est créée ou si celle-ci "disparaît" (parce qu'elle ne comprend plus au moins $f+1$ agents) avant la fin de sa traversée, aucun bon agent ne déclarera le rassemblement effectué au cours de cette phase. Dans le pire cas, les agents devront attendre jusqu'à faire la bonne hypothèse à propos de la configuration initiale, avant d'être témoins de la création d'une tour qui effectuera une traversée complète du graphe (et donc avant de déclarer que le rassemblement est effectué).

## Références

[BDD16] Sébastien BOUCHARD, Yoann DIEUDONNÉ et Bertrand DUCOURTHIAL. « Byzantine gathering in networks ». In : *Distributed Computing* 29.6 (2016), p. 435–457.

[CLP12] Jurek CZYZOWICZ, Arnaud LABOUREL et Andrzej PELC. « How to meet asynchronously (almost) everywhere ». In : *ACM Transactions on Algorithms* 8.4 (2012), p. 37.

[DPP14] Yoann DIEUDONNÉ, Andrzej PELC et David PELEG. « Gathering Despite Mischief ». In : *ACM Transactions on Algorithms* 11.1 (2014), p. 1.

[Rei05] Omer REINGOLD. « Undirected ST-connectivity in log-space ». In : *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, STOC*. 2005, p. 376–385.