Comment réduire efficacement l'entropie des sources malveillantes d'information

Silvia Bonomi, Jérémie Decouchant, Giovanni Farina, Vincent Rahli, Sébastien Tixeuil

Comment réduire efficacement l'entropie des sources malveillantes d'information[†]

Silvia Bonomi[1], Jérémie Decouchant[2], Giovanni Farina[1], Vincent Rahli[3], et Sébastien Tixeuil[4]

[1] *Sapienza University of Rome, Rome, Italy*
[2] *Delft University of Technology, Delft, The Netherlands*
[3] *University of Birmingham, Birmingham, UK*
[4] *Sorbonne Université, CNRS, LIP6, Paris, France*

Nous considérons un réseau (modélisé par un graphe) utilisé pour propager des informations. Dans ce contexte, une source d'information diffuse à l'ensemble du réseau un message. Si la source est fiable, c'est à dire qu'elle envoie le même message à tous ses voisins directs, on souhaite qu'un nombre limité de participants malveillants qui tentent de miner sa crédibilité en retransmettant des messages sourcés contradictoires, ne puisse pas berner les participants honnêtes. Si la source est malveillante (et qu'elle cherche à augmenter l'entropie en envoyant tout et son contraire à ses voisins directs), les participants honnêtes doivent diminuer l'entropie des messages issus de la source, soit en les ignorant, soit en délivrant un unique message (le même pour tous). Dans cet article, nous montrons que les méthodes dans la littérature pour résoudre ce problème peuvent être améliorées grâce à des optimisations spécifiques et inter-couches. Nos simulations montrent que ces optimisations peuvent être efficacement combinées pour diminuer la quantité totale d'informations transmises ou la latence du protocole.

Mots-clefs : Diffusion fiable, Tolérance aux fautes, Graphe incomplet

1 Introduction

Les systèmes distribués que nous considérons sont constitués d'entités (ou processus) autonomes qui *communiquent* pour résoudre globalement des tâches non triviales. Dans ce contexte, communiquer de manière fiable peut devenir difficile, en particulier lorsque les processus doivent s'appuyer sur d'autres (potentiellement défectueux, ou *Byzantins*) pour transmettre des informations (c'est-à-dire que le réseau de communication est partiellement connecté). Deux abstractions de communication globale utiles ont été définies dans ce contexte. Premièrement, la *communication fiable* (RC) exige que : *(i)* lorsqu'un processus correct diffuse un message, ce message est délivré par tous les processus corrects, et *(ii)* lorsqu'un message provenant d'un processus est délivré, il a bien été émis par ce processus. Deuxièmement, la *diffusion fiable* (RB) considère le cas supplémentaire où l'expéditeur d'un message peut être malveillant (*i.e.*, Byzantin) et envoyer des messages contradictoires. Dans ce cas, tous les processus corrects sont censés délivrer le même message. De nombreux protocoles tolérants aux pannes reposent sur une primitive de diffusion fiable, telle que le consensus Blockchain [YKDEV19] et les transferts de propriété [CGK+].

Bracha [Bra87] a décrit le premier protocole RB tolérant f Byzantins (BRB) pour des réseaux fiables, asynchrones et entièrement connectés. Ce protocole est caractérisé par trois phases de communication tous-vers-tous de messages (à savoir, SEND, ECHO et READY) et assure la progression de l'algorithme dès qu'un quorum suffisant de nœuds a reçu suffisament de messages d'un type donné. Concernant les réseaux où une connectivité complète ne peut être supposée, Dolev [Dol81] a montré que des processus corrects peuvent communiquer de manière fiable en présence de f processus malveillants (c'est-à-dire résoudre le problème de communication fiable avec des Byzantins, BRC) si, et seulement si, le graphe du réseau est $(2f+1)$-connecté. En particulier, l'algorithme de Dolev permet à un processus p_i de délivrer un message lorsqu'il

[†]Ce travail a été financé en partie par les projets ANR ESTATE, ref. ANR-16-CE25-0009-03, EURASIA, et *progetto ateneo CALYPSO*.

le reçoit via au moins $f+1$ chemins disjoints uniquement constitués de processus corrects (ce qui est rendu possible lorsque le message transite par au moins $2f+1$ chemins disjoints). Les solutions de Bracha et Dolev supposent une connexion point à point authentifiée et des liens de communication fiables. Bonomi et al. [BFT19] ont proposé plusieurs optimisations qui améliorent les performances de l'algorithme de Dolev sur des topologies inconnues. D'autres approches ont supposé des processus authentifiés (c'est-à-dire que les processus peuvent utiliser des signatures numériques) au lieu de canaux de communication authentifiés [CL02]. S'appuyer sur la cryptographie fournit des propriétés d'intégrité et d'authenticité qui simplifient les algorithmes. En particulier, une connectivité et des exigences système plus faibles sont nécessaires pour résoudre les problèmes de BRB et de BRC dans le cas authentifié. Cependant, la cryptographie a un coût de calcul et nécessite une infrastructure à clé publique (PKI) de confiance. D'un point de vue théorique, les approches basées sur la cryptographie sont limitées au cas d'adversaires bornés par calcul. Dans cet article, nous visons des solutions capables de faire face à des adversaires à la puissance de calcul illimitée.

Les protocoles de Bracha et Dolev peuvent clairement être composés pour résoudre le problème BRB sur un réseau de communication général. En effet, une solution au problème BRC permet de simuler un réseau de communication entièrement connecté de liens fiables et authentifiés où se déploient des protocoles nécessitant des communications tous-vers-tous. Ce travail vise à augmenter le couplage entre les deux solutions afin d'améliorer l'efficacité du protocole composé (qui est une solution au problème BRB). Les métriques de performance pour évaluer nos modifications sont la *latence*, c'est-à-dire le temps écoulé entre la diffusion d'un message via la primitive BRB et la dernière délivrance du message par un processus, et la *quantité d'informations échangées* sur le réseau (mesuré en bits). Plusieurs travaux visant le même objectif existent [GKM$^+$, NRS$^+$] mais, à notre connaissance, ils supposent tous un graphe de communication complet et/ou une infrastructure cryptographique. Nous fournissons ici une présentation informelle de la plupart des optimisations inter-couches que nous avons identifiées et une partie des expérimentations menées pour évaluer les améliorations proposées. Des preuves de corrections, divers détails d'implémentation et des expérimentations supplémentaires sont disponibles dans la version longue de cet article [BDF$^+$].

2 Modèle et Problème

Nous supposons un ensemble fixe de n processus, chacun étant doté d'un identifiant unique. Chaque processus peut échanger des messages avec un sous-ensemble d'autres processus, ses voisins, via les canaux de communication. Nous modélisons ces interactions possibles à travers un graphe $G(V, E)$ dans lequel les nœuds sont les processus et les liens correspondent aux canaux de communication disponibles. La topologie du graphe n'est pas connue des processus. Nous supposons que les messages ne sont pas perdus ou modifiés lors des échanges et que chaque processus ne peut pas mentir sur son identité lorsqu'il s'adresse à l'un de ses voisins (c'est-à-dire que nous supposons que les canaux de communication sont fiables et authentifiés). Nous supposons que dans le système, il ne peut y avoir qu'un nombre limité de processus, au plus f, qui peuvent avoir un comportement arbitraire (*pannes Byzantines*). Les autres processus sont corrects, c'est à dire qu'ils exécutent fiablement et honnêtement le code du protocole. Nous supposons que la nœud-connexité du réseau k est supérieure au double des processus fautifs f (c'est-à-dire $k > 2f$) et que plus des deux tiers des pairs sont corrects (c'est-à-dire $n > 3f$), conditions nécessaires à la résolution du problème analysé [Bra87, Dol81]. Les processus connaissent les identifiants de tous les processus et la valeur de f.

Notre objectif est de résoudre le problème de *diffusion fiable* dans le système considéré. Une solution à ce problème fournit deux interfaces, *diffuse* et *délivre*, qui garantissent les propriétés suivantes : *[BRB-Validité]* Si un processus correct (la source) diffuse m, alors tous les autres processus corrects délivrent m ; *[BRB-Pas de duplication]* Aucun processus correct ne délivre m plus d'une fois ; *[BRB-Integrité]* Si un processus correct délivre m d'expéditeur p_i, alors m a été précédemment diffusé par p_i ; *[BRB-Accord]* Si un processus correct délivre m, alors chaque processus correct délivre m.

3 Protocole inter-couches

Nous présentons plusieurs modifications visant à réduire la latence et la quantité de données de la combinaison des protocoles de Dolev et Bracha. Dans ce but, nous avons identifié plusieurs ajustements qui visent

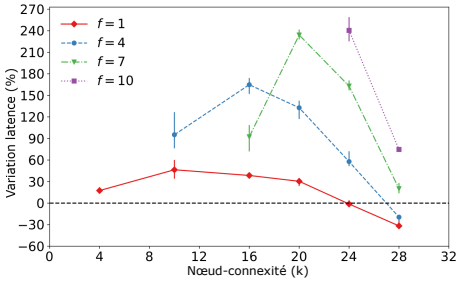

FIGURE 1 : MBD.11 : Impact sur la latence.

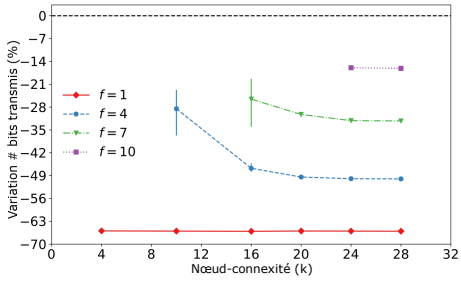

FIGURE 2 : MBD.11 : Impact sur la consommation réseau.

à *(i)* limiter le nombre de fois que le contenu utile d'un message est échangé, *(ii)* limiter les redondances de messages en agrégeant certains d'entre eux, et *(iii)* arrêter la propagation de messages inutiles pour l'avancement du protocole. La liste qui suit présente plusieurs des modifications décrites dans l'article long (en gardant les mêmes noms pour faciliter les correspondances).

[MBD.1] - Un processus p_i, lors de la réception d'un message dont le contenu est alors inconnu, lui associe un identifiant local unique. Lors du premier envoi d'un message à ses voisins lié à ce contenu, p_i inclut alors l'identifiant local choisi. Dans ses messages ultérieurs, p_i n'envoie que l'ID local de ce contenu.

[MBD.2] - Les messages SEND sont envoyés uniquement aux voisins de la source et ils ne sont pas propagés. En particulier, MBD.2 nécessite la modification suivante.

[ECHO amplification] - Si un processus RC-délivre $f + 1$ messages ECHO provenant de processus distincts, alors il envoie un message ECHO.

[MBD.3] - Un processus p_i qui n'a pas encore diffusé de message ECHO pour un contenu utile et qui RC-délivre un message ECHO de p_j génère un nouveau message ECHO_ECHO regroupant les messages ECHO de p_i et p_j.

[MBD.4] - Un processus p_i qui n'a pas encore diffusé de message READY pour un contenu utile et qui RC-délivre un message ECHO de p_j, au lieu de générer un message READY, génère un nouveau message READY_ECHO agrégeant le message ECHO de p_j et le message READY p_i.

[MBD.6] - p_i ignore les messages ECHO de p_j après que p_i a RC-délivré le message READY de p_j relatif au même contenu.

[MBD.7] - p_i ignore les messages ECHO liés à un contenu après sa BRB-délivrance.

[MBD.8] - Si p_i RC-délivre READY de p_j, alors il peut ignorer tout message ECHO qu'il reçoit de p_j.

[MBD.9] - Si p_i RC-délivre le message READY de $2f + 1$ de ses voisins, il peut arrêter de propager d'autres messages liés au même contenu utile à ces mêmes voisins.

[MBD.11] - Les processus avec les $\lceil \frac{n+f+1}{2} \rceil + f$ plus petits ID génèrent des messages ECHO, tandis que les processus avec les $2f+1+f$ plus petits ID génèrent des messages READY. Les autres processus relaient simplement les messages qu'ils reçoivent. Tous les processus BRB-délivrent livrent lorsqu'ils ont collecté $2f+1$ messages READY.

[MBD.12] - La source peut transmettre son message SEND à $2f+1$ de ses voisins, au lieu de tous.

Remarque : Il faut noter que nos modifications préservent la satisfaction des spécifications d'origine (cf. preuve dans [BDF+]), et nous nous focalisons sur leur impact pratique sur la latence et la charge réseau.

4 Évaluations expérimentales

Nous évaluons l'impact des modifications MBD.1–12 sur la combinaison des algorithmes de Bracha et Dolev incluant les améliorations de Bonomi et al. [BFT19] en utilisant le simulateur *Omnet++* [Omn]. On fait varier le nombre n de processus, le nombre f de processus Byzantins et la connectivité réseau k. Le code source de nos expériences est disponible en ligne [Cod].

Une partie des résultats que nous avons obtenus est résumée dans le tableau 1, où l'amélioration de la latence du protocole et de la complexité des données est rapportée pour un seul scénario sur un graphe régulier aléatoire de 31 nœuds. Il est possible de diminuer la quantité totale d'informations transmises ou la latence du protocole (par exemple, respectivement, -25% et -50% avec une charge utile de 16 octets, 31

TABLE 1 : Impact des modifications sur des topologies aléatoires de 31 nœuds avec des communications synchrones.

		Petit contenu utile (16o)				Gros contenu utile (16Ko)			
MBD	**Niveau**	**Lat. var. %**	**Utile quand**	**# bits var.**	**Utile quand**	**Lat. var. %**	**Utile quand**	**# bits var.**	**Utile quand**
1	Bra	$[-22, 4.3]$	toujours	-63	toujours	$[-93, -78]$	toujours	-97	toujours
2	BraDol	$[-21, 62]$	k élevé	$[-1.8, 9]$	f petit $\vee$ k élevé	$[-15.4, 107]$	k élevé	$[-4.3, 0.4]$	toujours
3	BraDol	$[-21, 99]$	k élevé	$[-1.8, 12]$	k élevé	$[-17, 104]$	k élevé	$[-4.3, 0.4]$	k élevé
4	BraDol	$[-25, 5]$	k élevé	$[-1.4, 19.7]$	f=1 $\vee$ k élevé	$[-50, 1.2]$	toujours	$[-1.3, 0.8]$	f=1 $\vee$ k élevé
6	Bra	$[-5.3, 1.6]$	-	$[-9.6, 0]$	toujours	$[-4, 4.5]$	-	$[-0.9, 0.13]$	toujours
7	Bra	$[-3, 1.4]$	-	$[-13.2, -1.4]$	toujours	$[-2, 5.4]$	-	$[-5.8, 0.07]$	toujours
8	Bra	$[-8.7, 3.8]$	-	$[-12.8, -3.1]$	toujours	$[-3.6, 2]$	-	$[-4.9, 0.07]$	toujours
9	BraDol	$[-5.7, 3.0]$	-	$[-43, 0]$	toujours	$[-3.9, 1.9]$	-	$[-38, 0]$	toujours
11	Bra	$[-25, 149]$	f petit $\wedge$ k élevé	$[-66, -16]$	toujours	$[-31, 240]$	f petit $\wedge$ k élevé	$[-66, -16]$	toujours
12	Bra	$[-15, 27]$	k élevé	$[-2.7, 2.6]$	f et k petits	$[-17, 57]$	k élevé	$[0.26, 4.7]$	jamais

participants dont 4 malveillants). Avec les mêmes paramètres, les Figures 2 et 1 montrent l'impact de la modification MBD.11 sur la latence du protocole et sur sa consommation réseau avec $f = 1, 4, 7, 10$. Cette modification réduit la consommation de bande passante sensiblement (jusqu'à -70%) quand la connectivité du réseau est faible mais augmente aussi la latence du protocole (jusqu'à +240%). L'ensemble des résultats est détaillée dans la version longue [BDF+].

5 Conclusion

Nous avons montré qu'une meilleure intégration entre les protocoles de Dolev et Bracha permet de résoudre le problème de diffusion fiable Byzantine sur un réseau de communication général avec une latence et une quantité de données échangées réduites par rapport à l'état de l'art. Nous envisageons plusieurs travaux futurs, tels que des règles d'agrégation plus générales tirant parti du fait que les processus sont à la fois relais et destination des messages échangés, et une extension au cas où la topologie du réseau est connue et à une évaluation sur des systèmes réels.

Références

[BDF+] Silvia Bonomi, Jérémie Decouchant, Giovanni Farina, Vincent Rahli, and Sébastien Tixeuil. Practical byzantine reliable broadcast on partially connected networks. In *ICDCS 2021*.

[BFT19] Silvia Bonomi, Giovanni Farina, and Sébastien Tixeuil. Multi-hop byzantine reliable broadcast with honest dealer made practical. *J. Braz. Comput. Soc.*, 25(1) :9 :1–9 :23, 2019.

[Bra87] Gabriel Bracha. Asynchronous byzantine agreement protocols. *Inf. Comput.*, 75(2) :130–143, 1987.

[CGK+] Daniel Collins, Rachid Guerraoui, Jovan Komatovic, Petr Kuznetsov, Matteo Monti, Matej Pavlovic, Yvonne-Anne Pignolet, Dragos-Adrian Seredinschi, Andrei Tonkikh, and Athanasios Xygkis. Online payments by merely broadcasting messages. In *DSN 2020*.

[CL02] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Trans. Comput. Syst.*, 20(4) :398–461, 2002.

[Cod] Code source, `https://github.com/jdecouchant/BRB-partially-connected-networks`.

[Dol81] Danny Dolev. Unanimity in an unknown and unreliable environment. In *FOCS*, 1981.

[GKM+] Rachid Guerraoui, Petr Kuznetsov, Matteo Monti, Matej Pavlovic, and Dragos-Adrian Seredinschi. Scalable byzantine reliable broadcast. In *DISC 2019*.

[NRS+] Kartik Nayak, Ling Ren, Elaine Shi, Nitin H. Vaidya, and Zhuolun Xiang. Improved extension protocols for byzantine broadcast and agreement. In *DISC 2020*.

[Omn] Omnet++ Discrete Event Simulator, `omnetpp.org`.

[YKDEV19] Jiangshan Yu, David Kozhaya, Jérémie Decouchant, and Paulo Jorge Esteves-Veríssimo. Repucoin : Your reputation is your power. *IEEE Trans. Computers*, 68(8) :1225–1237, 2019.