

Gestion des connaissances partagées par des agents à ressources et connectivité limitées : étude, analyse et expérimentation

M. Limame^a
mohamed.limame@etu.sorbonne-universite.fr

J. Henriët^b
julien.henriet@univ-fcomte.fr

C. Lang^b
christophe.lang@univ-fcomte.fr

N. Marilleau^a
nicolas.marilleau@ird.fr

^aInstitut de recherche pour le développement,
Sorbonne Université, France

^bInstitut FEMTO-ST,
université de Franche-Comté, France

Résumé

Un système distribué se compose d'un ensemble de nœuds connectés, communiquant et partageant des données. N'ayant pas accès à une horloge centrale, la cohérence de ces données partagées est l'une des problématiques importantes posées par ce type de système. Des approches permettant d'instaurer une cohérence temporelle de ces données partagées ont été conçues et implémentées par le passé. Cette problématique étant également rencontrée dans les systèmes multi-agents, notre étude présente et analyse les protocoles de gestion de la cohérence temporelle existants afin d'identifier ceux qui sont en adéquation avec les spécificités des systèmes multi-agents à ressources et connectivité limitées et faiblement accointants. Nous proposons dans cet article une classification et une analyse comparative de ces protocoles en vue d'une utilisation au sein de ses systèmes multi-agents. Nous allons aussi effectuer une simulation de la nouvelle approche smasdev, une des méthodes de synchronisation étudiées, pour interpréter ensuite les résultats obtenus.

Mots-clés : Gestion de la cohérence, cohérence temporelle, données partagées, synchronisation, systèmes multi-agents, systèmes multi-agents à ressources et connectivité limitées, smasdev

Abstract

A distributed system consists of a set of nodes connected, communicating and sharing data. Since it does not have access to a central clock, the consistency of this shared data is one of the important issues posed by this type of system. Approaches to establish temporal consistency of this shared data have been designed and implemented in the past. As this issue is also encountered in multi-agent systems, our study presents

and analyses existing protocols for managing temporal coherence in order to identify those that are in line with the specificities of weakly adjoining multi-agent systems with limited resources and connectivity. We propose in this paper a classification and a comparative analysis of these protocols for use within this category of multi-agent systems. We will also perform a simulation of the new smasdev approach, one of the studied synchronisation's methods, and then interpret the results obtained.

Keywords: Coherence management, temporal coherence, shared data, synchronisation, multi-agent systems, multi-agent systems with limited resources and connectivity, smasdev

1 Introduction

Au regard de l'importance que représente la notion temps dans les systèmes multi-agents dans l'interprétation des informations recueillies par chacun des agents et dans la détermination du comportement des agents notamment lors des échanges de données, nous nous intéressons dans cet article aux approches utilisées dans les systèmes distribués et dont l'objectif est d'assurer une cohérence temporelle des données partagées. Nous proposons dans cet article une classification et une analyse comparative des protocoles de la littérature relative aux systèmes distribués en vue d'une utilisation au sein d'un système multi-agents (SMA) à ressources et connectivité limitées.

Selon B. Chaib-Draa [2], un SMA est un système distribué composé d'éléments particuliers appelés agents qui partagent des connaissances et communiquent entre eux afin de résoudre un problème qui dépasse la portée d'un seul agent.

De ce fait, la problématique de la gestion de la cohérence temporelle s'impose également aux SMA, notamment dans les domaines d'applications sensibles comme la surveillance. En effet, il est nécessaire que les informations recueillies par chacun des agents pris individuellement soient cohérentes avec celles des autres agents du SMA pour que l'objectif global du système soit atteint. Comme précisé, nous nous focalisons dans cette étude sur les SMA à ressources et connectivité limitées qui se caractérisent par une faible capacité de calcul, une faible capacité de stockage, une énergie limitée et notamment une activité réduite focalisée sur sa survie avec un faible besoin en connectivité.

Selon A. Tanenbaum [28], un système distribué est un ensemble d'ordinateurs autonomes reliés par un réseau informatique qui apparaît à ses utilisateurs comme un seul système cohérent. Ces ordinateurs partagent des informations et des ressources sur une large étendue géographique ce qui a donné lieu dans certaines applications à des problèmes de cohérence temporelle notamment dans les applications temps réel utilisées dans les usines, l'aéronautique, les véhicules spatiaux et les applications militaires et également plus largement dans la gestion des versions et le contrôle des accès concurrents dans les systèmes distribués de base de données. Dans les systèmes distribués plusieurs approches ont été identifiées pour pallier à cette problématique. Il est impératif de comprendre les caractéristiques des agents pour distinguer parmi les approches utilisées dans les systèmes distribués celles qui sont en adéquation avec leurs besoins et contraintes.

Les principales contributions de cet article sont une classification des approches et algorithmes de synchronisation les plus utilisés dans les systèmes distribués et une expérimentation de la nouvelle approche smasdev. Le travail de classification et les résultats d'expérimentation de smasdev aideront ainsi les chercheurs dans l'établissement du choix de la technique de synchronisation la plus en adéquation avec les contraintes de leur propre système multi-agents faiblement accointants à ressources et connectivité limitées.

Dans ce qui suit, nous allons dans un premier temps définir la cohérence pour passer ensuite à la définition de la cohérence temporelle. Nous passons ensuite à la présentation des approches de synchronisation qui seront étudiées dans cet article et qui permettent d'instaurer une cohérence temporelle au sein d'un système multi-agents. Puis nous définissons les critères de clas-

sification avant de présenter la qualification des approches selon ces critères. Nous passons ensuite par une analyse comparative des approches citées. Nous terminerons par une simulation de l'approche smasdev et une interprétation des résultats obtenus.

2 Cohérence : concept et méthodes

2.1 Définitions de la cohérence

Il existe plusieurs définitions de la cohérence dans la littérature comme le confirme M. Olsson [21]. DJW Strümpfer [27] a défini la cohérence comme un moyen de mesure de l'absence de concept d'affectivité négative qui est similaire au névrosisme. En linguistique, selon B. Grosz et CL. Sidner [3] et K. Tracy et RT. Craig [9], un discours (texte ou dialogue) peut être qualifié de cohérent lorsque ses parties "vont ensemble" et structurent ainsi le discours. P. Thagard [29] [30] apporte une vision épistémologique en définissant la cohérence comme étant une théorie cognitive traduisant un niveau de satisfaction des contraintes contenues dans la connaissance individuelle sous ses différentes formes : explicative, analogique, déductive, perceptuelle et conceptuelle.

Dans les systèmes distribués, selon J. Sablatnig *et al.* [13] le concept de cohérence se traduit par la conservation d'une logique notamment temporelle, explicative ou conceptuelle assurée par une algorithmique et des contraintes. Sa gestion s'avère essentielle pour éviter les erreurs de calcul, d'une part, et les anomalies d'exécution des algorithmes d'autre part.

Dans les systèmes multi-agents, selon Qan. Vu [31] la cohérence des informations partagées dans le système est définie par la compatibilité des informations recueillies et stockées par chacun des agents. Contrairement à la cohérence dans les systèmes distribués où celle-ci est souvent globale et associée à l'ensemble du système. La cohérence des données partagées dans les systèmes agents est plutôt émergente et résulte des cohérences individuelles.

La cohérence temporelle dans un système distribué a été introduite par J-P. Thomesse [14] lors des travaux de recherche sur les réseaux de terrain et est relative à la notion de « fenêtre temporelle ». Pour ce dernier, la notion de cohérence temporelle s'applique à un ensemble d'informations et vise à garantir que les informations appartenant à cet ensemble ont été pro-

duites dans une même « fenêtre temporelle ». Selon L. Cauffriez et J. Defrenne [1], deux types de cohérence temporelle peuvent être distingués dans les systèmes répartis par rapport au niveau d'exigence de l'exactitude de la notion temps entre les différents éléments du système multi-agents. Une cohérence temporelle stricte dans laquelle tous les éléments et connaissances détenus par l'ensemble des agents ont exactement le même estampille temps et une cohérence temporelle non stricte dans laquelle des différences dans l'estampillage des éléments et connaissances peuvent exister entre les agents. La cohérence temporelle au niveau global est atteinte lorsque la cohérence temporelle de chaque élément du système multi-agents est respectée.

2.2 Méthodes

Parmi les outils et méthodes pouvant être appliquées pour instaurer une cohérence temporelle, il y a la technique de synchronisation de la notion temps entre agents d'un SMA par l'utilisation d'un protocole de synchronisation, il y a aussi selon J. Ferber [12], la technique de l'**arbitrage** qui conduit à la définition de règles de comportement qui agissent comme des contraintes sur l'ensemble des agents. J. Ferber [12] évoque également l'élaboration d'une **convention de priorité** ou l'établissement d'une **procédure de vote et élection**. Selon S. Bussmann et H.J. Muller [24], il existe aussi la **négociation** qui permet d'atteindre un accord mutuellement accepté entre agents.

Différents protocoles de synchronisation sont basés sur des horloges physiques et logiques. Une **horloge physique** est un processus physique couplé à une méthode de mesure du temps. La plupart des horloges physiques sont basées sur des processus cycliques basés sur un oscillateur et un compteur. Une **horloge logique** quant à elle, est un moyen d'horodatage qui recense les relations chronologiques et causales dans un système distribué pour établir un classement global des événements de différents processus du système.

Algorithmes de synchronisation d'horloges physique.

Le concept de synchronisation se base sur une source d'horloge physique dont la donnée se propage dans un réseau. Nous allons nous intéresser dans notre étude au protocole NTP, à l'algorithme de Cristian et à l'algorithme de Berkeley.

Le **protocole NTP** utilise un système hiérarchique semi-stratifié de sources de temps se basant sur l'UTC (*coordinated universal time*) comme temps de référence. Comme décrit par N. Minar [20], chaque niveau de cette hiérarchie est appelé *strate* et se voit attribuer un numéro. Un serveur synchronisé sur un serveur de strate n s'exécute à la strate $n + 1$.

L'**algorithme de Cristian** selon [4] est un algorithme de synchronisation d'horloges utilisé pour synchroniser l'horloge locale d'un élément du système distribué avec un serveur de temps externe distant. Un élément du système envoie une demande au serveur temps pour recevoir l'heure actualisée.

L'**algorithme de Berkeley** est décrit par R. Gussella et S. Zatti [10]. Contrairement à l'algorithme de Cristian, le serveur de temps est actif. Il interroge périodiquement chaque élément du système distribué pour lui demander son horloge locale. Sur la base des réponses obtenues, il calcule une durée moyenne et demande à tous les éléments du système d'avancer ou ralentir leurs horloges vers la nouvelle horloge calculée.

L'implémentation dans un SMA du protocole NTP nécessite la mise en place d'une connexion inter-agents respectant une architecture NTP pour l'envoi de la donnée temps et requiert une connectivité continue des agents de la strate $n1$ à une source de temps. Que ce soit pour l'implémentation dans un SMA de l'algorithme de Berkeley ou celui de Cristian, il faut sélectionner un agent du SMA comme étant agent serveur et assurer une connectivité continue de cet agent avec l'ensemble des autres agents pour le déroulement de la synchronisation.

Algorithmes de synchronisation d'horloges logiques.

Le concept d'horloge logique a été initialement proposé par L. Lamport en 1978 [15]. Plusieurs auteurs ont proposé l'utilisation des horloges logiques pour détecter la relation de précedence causale entre les événements (F. Mattern *et al.* [19], J. Fidge [7], R. Schwarz et F. Mattern [25]).

L'**horloge de Lamport** selon [15] démontre que la synchronisation ne doit pas forcément être absolue et qu'elle peut être déduite en se basant sur les relations entre événements. L'ordonnement proposé permet d'attribuer une horloge logique ou estampille à tous les événements d'un système distribué.

Avec l'**horloge de Mattern**, selon F. Mattern *et*

al. [19], chaque élément p du système distribué possède un vecteur d'entiers appelé estampille dans lequel chaque composant $estampille[i]$ est l'estimation par p de la valeur de l'horloge de Lamport du processus i .

Avec les **horloges matricielles** présentées par M. Raynal et M. Singhal [23], chaque processus p d'un système distribué de n processus possède une matrice $n \times n$ d'estampilles dans laquelle chaque composant $estampille[i]$ est l'estimation par p de la valeur de l'horloge de Mattern du processus i .

Pour les 3 approches, une implémentation dans un SMA traduit le principe d'échange entre processus par un échange entre agents. Le ou les estampilles doivent être envoyés lors des interactions inter-agents pour assurer une synchronisation.

Synchronisation d'horloges dans les réseaux de capteurs.

Diverses applications utilisant les réseaux de capteurs sans fil notamment dans la surveillance et la fusion de données, nécessitent que tous les nœuds aient des horloges synchronisées. Or, les méthodes de synchronisation appliquées traditionnellement dans l'industrie ne sont pas forcément adaptées pour une utilisation dans les réseaux de capteurs en raison notamment de problèmes liés à la consommation d'énergie. Par exemple, le protocole NTP, bien que largement utilisé pour la synchronisation d'horloge sur Internet, n'est pas adapté à une utilisation dans les réseaux sans fil, car trop gourmand en énergie. De nouvelles approches adaptées à une utilisation dans ce type de réseau ont ainsi vu le jour ces dernières années.

Protocole de synchronisation de diffusion de référence (RBS) : Ce protocole est basé sur un schéma de synchronisation récepteur-récepteur. Comme décrit par J. Elson *et al.* [6], chaque nœud synchronise son horloge locale avec toutes les autres horloges des nœuds se trouvant dans sa portée de transmission.

Le **protocole de synchronisation temps pour les réseaux de capteurs (TPSN)** est basé sur un schéma de synchronisation émetteur-récepteur. L'algorithme décrit par S.Ganeriwala *et al.* [8] fonctionne en deux étapes. Au cours de la première étape, une structure hiérarchique est établie dans le réseau puis vient une seconde étape de synchronisation durant laquelle les nœuds d'un niveau i sont synchronisés avec les nœuds du niveau $i - 1$ par paire.

Le **protocole de synchronisation temps par inondation (FTSP)** décrit par M.Maróti *et al.* [18] met en œuvre une synchronisation dans laquelle le nœud racine transmet périodiquement un message de synchronisation unique aux nœuds qui sont à portée de transmission.

La **mesure du retard de synchronisation du temps pour les réseaux de capteurs sans fil (DMTS)** repose sur une synchronisation émetteur-récepteur dans laquelle l'expéditeur et plusieurs récepteurs sont synchroniser en même temps comme présenté par S. Ping [22].

La **technique de synchronisation temporelle consensuel (CCS)** vise à réduire les écarts d'horloges entre les nœuds proches et fait converger l'ensemble des nœuds vers un écart en commun. L'idée principale de cet algorithme selon M.K. Maggs *et al.* [17] est de compenser sur plusieurs itérations les écarts d'horloge entre les nœuds du système.

Le **protocole de synchronisation gradient (GTSP)** a été conçu par P. Sommer et W.Roger [26]. Dans ce protocole, les nœuds du réseau diffusent périodiquement un *beacon* de synchronisation avec leurs nœuds voisins. L'horloge logique de chaque nœud converge vers une horloge logique commune.

L'**algorithme reachback firefly (RFA)** cette approche se base sur un réseau d'oscillateurs à couplage d'impulsions où chaque nœud est un oscillateur qui émet périodiquement une impulsion auto-générée. En observant les impulsions des autres oscillateurs, un nœud ajuste légèrement la phase de son propre oscillateur.

Les réseaux de capteurs sans fil pouvant être considéré comme étant des SMA, l'implémentation de ces protocoles dans un SMA reprendra les traitements effectués ou subits au niveau des nœuds pour les appliquer sur les agents du SMA.

Synchronisation utilisant les données.

La **synchronisation des SMA basée sur l'évolution des données (SMASDEV)** selon M. Límame *et al.* [16] s'appuie sur le contenu d'un événement et l'évolution de la donnée qui lui est associée pour trier les événements. Cette nouvelle approche a pour objectif d'instaurer une cohérence globale au niveau d'un SMA en permettant à chaque agent de rétablir un ordre chronologique des données qu'il reçoit des autres agents, sur la base de ses connaissances et sans faire appel à une horloge (physique ou logique). Le prin-

cipe consiste à ce que chaque agent du système enregistre en mémoire sa perception personnelle de l'évolution d'une donnée qu'il recueille avec une fréquence déterminée par rapport à son horloge locale. L'agent peut ainsi prévoir l'évolution future de la donnée. Ainsi, l'agent doit disposer d'un modèle d'évolution en adéquation avec la donnée en question. Pour passer d'une perception personnelle vers une perception globale, chaque agent du système doit, dans un premier temps, demander aux autres agents de lui transmettre les données recueillies puis dans un deuxième temps les positionner par rapport à sa perception personnelle. Ainsi l'agent sera en mesure d'identifier le positionnement de l'ensemble des données reçues par rapport à son horloge. En conséquence, les agents du système seront synchrones et en phase par rapport au modèle d'évolution de la donnée dans le temps.

Selon J. Henriot [11], le **Pèlerin optimiste** est un protocole de gestion de la cohérence des données d'une mémoire partagée répartie de type mémoire répliquée à partage d'objets. Il fonctionne sur la base d'un jeton circulant sur un anneau logique unidirectionnel contenant une structure de données avec les mises à jour des données partagées.

3 Classification des méthodes avec des critères d'évaluation

3.1 Définition des critères

Pour une mesure efficace de la pertinence d'une approche de synchronisation par rapport à une autre, les critères de classification choisis doivent être suffisamment caractérisants. Nous distinguons deux familles de critères.

La famille des critères "*détails de la synchronisation*" permet de qualifier le fonctionnement des approches de synchronisation et le degré de synchronisation obtenu.

La famille des critères "*énergie*" permet de qualifier les besoins des approches de synchronisation notamment en mémoire et puissance pour pouvoir s'exécuter. Les critères de cette famille permettent d'estimer l'énergie requise par l'approche de synchronisation.

Famille des critères "*détails de la synchronisation*" :

Le *principe de synchronisation* permet de préciser le mécanisme de synchronisation utilisé par l'approche.

Le *type de cohérence* permet de traduire le niveau de synchronisation dans le réseau entre les agents. Dans la littérature il existe plusieurs niveaux de synchronisation, nous en considérons les quatre identifiés par L. Cudennec [5] :

- Le **niveau de cohérence atomique** traduit un degré de synchronisation global stricte au sein de l'ensemble des agents du réseau.
- Le **niveau de cohérence séquentielle** traduit la vision de L. Lamport dont la synchronisation se focalise sur l'ordre des événements en non sur le temps d'occurrence de l'événement.
- Le **niveau de cohérence causale** se base sur la relation de causalité entre les événements survenus dans le réseau d'agents.
- Le **niveau de cohérence relâchée** traduit le degré de synchronisation le plus faible par rapport à ceux cités précédemment.

L'*objet de synchronisation* précise la nature du ou des éléments faisant objet de synchronisation. La cohérence obtenue à travers la synchronisation est une cohérence temporelle si l'élément synchronisé est une donnée "temps".

Le *type de synchronisation* qualifie le processus de synchronisation par rapport à son déclenchement, c'est-à-dire s'il est systématique ou non systématique. Le critère de *tolérance aux pannes* permet d'indiquer si l'approche de synchronisation supporte la présence d'une ou plusieurs défaillances au niveau de la flotte d'agents.

Famille des critères "*énergie*" :

Le critère *calcul réalisé* permet de caractériser les opérations requises pour pouvoir lancer l'approche de synchronisation.

Le *volume échangé* permet de caractériser pour chaque approche de synchronisation le volume requis échangé entre deux agents pour assurer une synchronisation.

Le *nombre de messages requis pour synchroniser une donnée entre 2 agents* recense le nombre de messages requis échangés par deux agents pour assurer une synchronisation.

Le *nombre de messages requis pour synchroniser une donnée entre N agents* recense le nombre de messages requis échangés par un agent avec les autres agents pour assurer la synchronisation d'une donnée. Le nombre de messages échangés permet de quantifier l'énergie consommée pour l'exécution de l'approche de synchronisation au niveau de la flotte d'agents.

3.2 Classification des approches de synchronisation étudiées

Notre classification est présentée dans le tableau 1. Cinq des quinze approches considérées dans cet article supportent par défaut la tolérance aux pannes. Même si le reste des approches ne présente pas cette faculté, elle peut être mise en place lors de leur implémentation. Qu'il s'agisse des approches de synchronisation se basant sur un échange d'horloges ou de données, la topologie du réseau ne constitue pas un frein pour le lancement de l'approche de synchronisation. En effet, dans treize des quinze approches étudiées, les agents doivent pouvoir communiquer les uns avec les autres indépendamment de la topologie. Toutes les approches étudiées se basant sur les horloges physiques ou logiques assurent une synchronisation entre deux éléments à travers un envoi d'un à trois messages. Dans les approches utilisées dans les réseaux de capteurs sans fil, une synchronisation entre deux nœuds peut être réalisée avec un ou deux messages. Nous constatons que pour les approches RFA et DMTS, le nombre de messages envoyés pour la synchronisation globale d'un système multi-agents ne varie pas en augmentant ou en diminuant le nombre d'agents. En effet, l'émetteur envoie un seul message en mode diffusion vers l'ensemble des nœuds du réseau constitué d'agents.

Toutes les approches de synchronisation à l'exception de celles qui se basent sur les données nécessitent un espace mémoire relativement faible pour instaurer une cohérence temporelle. Le besoin en stockage pour les approches se basant sur les données dépend de la taille de l'objet de synchronisation et du résultat de la fonction utilisée pour la synchronisation. Parmi les approches étudiées, seul le *Pèlerin optimiste* permet d'assurer une cohérence atomique. Neuf des quinze approches étudiées, indépendamment de leurs domaines d'application, permettent d'effectuer une synchronisation de données en plus de la synchronisation de la notion temps (estampille, horloge,...). L'ensemble des approches de synchronisation étudiées se basant sur des échanges d'horloges physiques et logiques, correspondent à des approches à déclenchement systématique à l'occurrence d'événements. Par ailleurs, dans les réseaux de capteur sans fil, parmi les sept approches considérées, les approches RBS, TPSN et DMTS ne mettent pas en œuvre un déclenchement systématique.

4 Analyse et interprétations des résultats de classification

Cette classification constitue une aide à l'identification de la technique de synchronisation la plus adaptée à des contraintes d'application de SMA. Dans le cas d'agents ayant des ressources limitées, leur capacité de calcul peut être elle aussi limitée, l'espace mémoire réduit et les échanges de messages peuvent être restreints. D'une manière générale, plus les besoins de l'approche de synchronisation sont réduits, plus la durée d'application de l'approche est importante. Ce détail peut s'avérer crucial notamment dans les cas d'application sur des trajets et parcours comme lors de missions de cartographie menées par plusieurs agents. Cette interprétation est affinée dans la partie suivante ressource par ressource.

Concernant l'espace mémoire, l'approche de synchronisation se basant sur les Horloges matricielles se distingue par un besoin en espace mémoire important. En effet, le stockage de l'ensemble des matrices associées à chaque événement (local ou reçu) est coûteux. De même, à un degré moins important, la synchronisation utilisant les Horloges de Mattern a un besoin en espace mémoire relativement coûteux. L'utilisation de ces approches peut avoir un impact négatif sur la robustesse du système. A contrario, les approches DMTS, NTP, TPSN et l'algorithme de Cristian se caractérisent par un besoin relativement faible en espace de mémoire. Le besoin consiste essentiellement à stocker au niveau de l'agent récepteur l'horloge ou l'estampille temps à synchroniser. L'Horloge de Lamport, l'Algorithme de Berkeley, RBS, FTSP, CCS et le protocole du Pèlerin optimiste requièrent une faible capacité de calcul réduite à la puissance nécessaire au calcul d'une moyenne de temps, de l'offset ou de l'écriture sur un jeton.

Les algorithmes de Cristian et de Berkeley sont deux approches adaptées à des SMA dans lesquelles il est possible de gérer le temps de manière centralisée.

Sur le plan de la consommation d'énergie, c'est l'approche de synchronisation CCS qui nécessite le plus grand nombre de messages car en effet, l'envoi de messages à l'ensemble des agents du réseau se fait sur plusieurs itérations jusqu'à la suppression des écarts. Ceci peut s'avérer un facteur direct d'épuisement de l'énergie de l'agent. De plus, cette synchronisation est systématique la rendant d'autant plus coûteuse en énergie. De

TABLE 1 – " Classification des approches de synchronisation"

libellé	Famille des critères "détails de la synchronisation"					Famille des critères "énergie"			
	principe de synch.	type de cohérence	objet de synch.	Syncho systématique	tolérance aux pannes	Calcul réalisé	Volume échangé	nbr msg synch 2 agents	nbr msg synch n agents
Algorithmes de synchronisation d'horloges physiques									
Protocole NTP	partage d'une estampille	R	estampille	X	X	op soustraction + op division	une estampille	2	n-m+1 n niveau récepteur , m niveau émetteur
Algo de Cristian	à travers un serveur temps	S	horloge	X		op addition + op division	Round-trip + horloge serveur temps	2	2n
Algo Berkeley	à travers un serveur temps	R	Offset	X	X	op addition + op division	serveur : n horloges	3	3n
Algorithmes de synchronisation d'horloges logiques									
Horloges Lamport	partage d'une estampille	C	donnée + estampille	X		op comparaison + op incrémentation	estampille locale et synch	1	n-1
Horloges de Mattern	partage d'un vec. estampilles	S	vecteur d'estampilles	X		n-1 op comparaison + op incrémentation estampille	vecteur d'estampilles locales et synch	1	n-1
Horloges matricielles	partage d'une mat. estampilles	S	matrice d'estampilles	X		n-1 op comparaison + op incrémentation estampille	matrices d'estampilles locales et synch	1	n-1
Synchronisation d'horloges dans les réseaux de capteurs									
RBS	partage d'horloge	R	émetteur : tt donnée ; récepteur : horloge			op addition + op division + calcul offset	donnée partagée + k horloges(k nbr voisins)	2	1+(n-1)* k k : nbr noeux voisins aux (n-1) noeux ayant reçu l'horloge
TPSN	partage d'une estampille	R	paquet de synch.			mise en arborescence + calcul nouvelle horloge	une estampille	2	n-m+1 avec n niveau récepteur et m niveau émetteur
FTSP	partage d'horloge	R	ID + estampille racine + donnée	X	X	calcul offset	estampille + donnée partagée	2	1+k :1msg émetteur + 1msg récepteurs vers voisinage
DMTS	partage d'une estampille	R	estampille racine			calcul nouvelle horloge	une estampille	1	de l'émetteur vers récepteurs
CCS	partage d'une estampille	R	estampille et taux d'écart	X	X	calcul taux d'écart	estampille + taux d'écart	2 * nbr itérations	2*(n-1) * nbr itérations
GTSP	partage d'horloge	R	balise de synch et fréq d'horloge	X	X	mise à jour fréq d'horloge + calcul nouvelle horloge	horloge + fréquence d'horloge	2	2(n-1)
RFA	partage d'un état	S	vecteur d'état	X		calcul vecteur d'état et de la fonction objectif	packet synchronisation vecteur d'état	1	de l'émetteur vers récepteurs
Synchronisation utilisant les données									
Smasdev	assimilation de données	C	donnée ayant 1 modèle d'évol			manipulation donnée + simulation modèle d'évolution	tuplet de données partagées	1	n-1
Pélerin optimiste	Réservation ressources et diffusion via jeton	A	Données partagées			manipulation jeton	données partagées	6	3n+1

même, les protocoles NTP, FTSP, l'algorithme de Cristian, l'algorithme de Berkeley étant exécutés d'une façon régulière et périodique, ils risquent de consommer beaucoup d'énergie si la fréquence de lancement est importante. Quant aux approches qui se lancent à l'occurrence d'un événement ou l'une de leurs évolutions (RFA ou Horloges de Mattern par exemple), la consommation d'énergie augmente de manière modérée lors de chaque lancement ce qui est en adéquation avec une utilisation dans un système multi-agents à ressources et connectivité limitées. La consommation d'énergie pour les approches Kalman et Smasdev pour lesquelles la synchronisation n'est pas systématique, reste relativement faible et augmente proportionnellement avec la complexité du modèle d'évolution de la donnée partagée et la densité de répartition du réseau d'agents.

Bien que les sept approches étudiées et utilisées dans les réseaux de capteurs sans fil puissent être appliquées dans les SMA, il serait souhaitable de limiter au maximum la fréquence de leur exécution dans un système dans lequel les agents ont des ressources limitées.

En se basant sur l'approche de Lamport, la comparaison des valeurs des horloges ne permet pas de déduire une relation de causalité entre deux événements a et b . En effet, si $H(a) < H(b)$ ne signifie pas forcément que $a \Rightarrow b$. De plus, en cas d'affluence, des messages peuvent ne pas être reçus. Au regard de ces limites, nous proposons de l'écartier de la liste des approches retenues.

Le choix de l'approche de synchronisation peut aussi être déterminé en fonction de la nature de la donnée partagée. Par exemple, si la donnée partagée porte uniquement sur la notion de temps (estampille, matrice temps, ..), le choix peut être l'une des approches TPSN, DMTS, CCS ou GTSP conçues pour instaurer une synchronisation temporelle et en adéquation avec le type de système multi-agents étudié. Par contre si la donnée partagée ne porte pas sur la notion temps, nous proposons l'utilisation des approches RFA, Smasdev, RBS ou FTSP qui sont en adéquation avec le systèmes multi-agents étudié et qui sont triées selon l'ordre de degré de cohérence du plus strict vers le moins strict. Nous écartons de cette liste le Pèlerin optimiste nécessitant que tous les agents puissent communiquer continuellement et donc peu compatible avec un SMA constitué d'agents faiblement accointants.

Quoiqu'il en soit, les protocoles nécessitant le moins de ressources sont à privilégier pour

les SMA constitués d'agents à ressources et connectivités limitées. Au-delà de ces considérations, l'approche de synchronisation à choisir dépend fortement du contexte de synchronisation et du niveau de synchronisation recherché. Par exemple, pour répondre à une mission de supervision des données de température sur une zone à risques volcanique, l'approche smasdev semble convenir parfaitement. Nous allons dans ce qui suit effectuer une simulation sous la plateforme Gama de l'approche smasdev afin de vérifier les comportements obtenus.

5 smasdev : simulation et interprétation des résultats

5.1 Détails de simulation

Pour répondre à la mission de supervision des données de température sur une zone à risques volcaniques, nous allons tout d'abord implémenter un modèle agent doté d'une fonctionnalité de synchronisation basée sur smasdev. Pour que la simulation soit proche des conditions réelles d'exécution, nous avons ajouté les paramètres suivants :

Le *taux de connectivité* permet de prendre en considération les problèmes de connectivité qui peuvent survenir durant une mission. Elle est entre 0 et 1.

La *déviaton d'horloge* permet de prévoir les déviations en millisecondes que peut avoir une horloge notamment à cause des conditions climatiques extrêmes.

La *période de prise de mesures* permet de renseigner l'intervalle de temps en millisecondes entre deux prises de températures et ceci pour l'ensemble des agents durant leur mission.

En plus de ces paramètres, nous allons aussi faire varier le nombre d'agents mandatés pour la mission afin d'analyser l'impact de la taille de la flotte sur les résultats de synchronisation.

Pour tester le comportement de l'approche smasdev, nous allons utiliser 5 instances de simulations sous la plateforme gama dans sa version 1.8. Le SMA étudié étant faiblement connecté, le paramètre taux de connectivité sera positionné à 10%. La mission se déroulant sur une zone volcanique à haute température, le paramètre déviaton d'horloge sera positionné à 550ms (0.55s). Le dernier paramètre fixant l'intervalle de prise de mesure sera positionné à 50ms. Nous varions le nombre d'agents à chaque simulation.

Les données utilisées sont représentées dans le tableau ci-après :

	sim 1	sim 2	sim 3	sim 4	sim 5
nbr agents	3	5	10	15	20
Autres paramètres pour toutes les simulations :					
Déviation horloge					0.55s
Période prise de mesure					50ms
Taux de connectivité					10%

Nous utilisons, lors de la simulation, un échantillon de 190 données de température qui seront réparties aléatoirement entre les agents de la flotte en respectant l'intervalle de prise de mesure. Pour des raisons de simplification nous allons considérer que le modèle d'évolution de ces données suit une fonction de non régression linéaire de forme $y : ax+b$.

Lorsqu'un agent est connecté à un autre, il identifie, parmi les données de températures de l'autre agent, celles qui ne se trouvent pas dans sa base de données et lance ensuite l'approche de synchronisation smasdev pour les estampiller selon son horloge local. Le nombre de fois où un agent se trouve connecté à un autre dépend du paramètre "taux de connectivité".

La simulation prend fin au bout de 1500 pas d'horloge.

5.2 Interprétation des résultats de simulation

Une donnée correctement synchronisée par un agent est une donnée dont le positionnement par rapport à ses autres données recueillies est en adéquation avec le modèle d'évolution de la donnée.

Bien que le taux de connectivité soit faible, nous avons obtenu, lors des 5 simulations, un pourcentage de bon positionnement à 100% pour chaque agent. Le pourcentage est défini par le ratio du nombre de données correctement positionnées sur le nombre total de données synchronisées par l'agent. Smasdev est bien adapté à notre SMA étudié par rapport à la contrainte de connectivité.

Après avoir testé smasdev à une petite échelle, nous souhaitons valider l'approche smasdev avec un plan d'expérimentation plus riche et plus conséquent à travers une simulation en masse. Ceci peut être réalisé avec l'utilisation de la bibliothèque gamar.

6 Conclusion

Cet article présente une étude de différentes solutions envisageables à la problématique de la synchronisation temporelle des données partagées entre agents faiblement accointants et à ressources limitées.

La démarche consiste à partir des solutions existantes dans l'univers des systèmes distribués pour identifier celles qui sont en adéquation avec les contraintes du système multi-agents étudié notamment celle de la limitation des ressources.

Nous avons effectué une simulation de l'approche smasdev qui fait partie des approches étudiées pour vérifier sa compatibilité avec notre SMA étudié du point de vue de la connectivité.

Au regard de cette étude et des résultats de simulations, nous envisageons maintenant de valider que l'approche smasdev est complètement compatible avec notre SMA étudié en tenant compte de toutes les contraintes notamment celles liées à la consommation d'énergie. Pour cela nous allons effectuer une simulation à grande échelle tenant en compte de l'ensemble des paramètres pouvant influencer le comportement de smasdev.

Références

- [1] Laurent Cauffriez and Jean Defrenne. Viabilité de l'information et réseau à diffusion. *Diagnostic et sûreté de fonctionnement*, Vol, 5(2) :219–247, 1995.
- [2] Brahim Chaib-draa. Agents et systèmes multi-agents. 1999.
- [3] Robert T Craig and Karen Tracy. *Conversational coherence : Form, structure, and strategy*, volume 2. SAGE Publications, Incorporated, 1983.
- [4] Flaviu Cristian. Probabilistic clock synchronization. *Distributed computing*, 3 :146–158, 1989.
- [5] Loïc Cudennec. Modèles et protocoles de cohérence des données en environnement volatil. 2005.
- [6] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 36(SI) :147–163, 2002.
- [7] Colin J Fidge. Timestamps in message-passing systems that preserve the partial ordering. 1987.

- [8] Saurabh Ganeriwal, Ram Kumar, and Mani B Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 138–149, 2003.
- [9] Barbara Grosz and Candace L Sidner. Attention, intentions, and the structure of discourse. *Computational linguistics*, 1986.
- [10] Riccardo Gusella and Stefano Zatti. The accuracy of the clock synchronization achieved by tempo in Berkeley Unix 4.3 BSD. *IEEE transactions on Software Engineering*, 15(7) :847–853, 1989.
- [11] Julien Henriët. *Evaluation, optimisation et validation de protocoles distribués de gestion de la concurrence pour les interactions coopératives*. PhD thesis, 2005.
- [12] FERBER Jacques. Les systèmes multi-agents : vers une intelligence collective. *InterEditions, Paris*, 322, 1995.
- [13] Sablatnig Jan, Grottke Sven, Kopke Andreas, Chen Jiehua, Seiler Ruedi, and Wolisz Adam. Consistency in distributed systems. 2007.
- [14] Thomesse Jean-Pierre. Le réseau de terrain fip. (3) :287–321, 1993.
- [15] Leslie Lamport. Time, clocks and the ordering of events in a distributed system. *Communications of the ACM*, 1978.
- [16] Mohamed Limame, Julien Henriët, Christophe Lang, and Nicolas Marilleau. Synchronisation d’horloge dans un système multi-agents. In *APIA*, 2019.
- [17] Michael Kevin Maggs, Steven G O’keefe, and David Victor Thiel. Consensus clock synchronization for wireless sensor networks. *IEEE sensors Journal*, 12(6) :2269–2277, 2012.
- [18] Miklós Maróti, Branislav Kusy, Gyula Simon, and Akos Lédeczi. The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 39–49, 2004.
- [19] Friedemann Mattern et al. *Virtual time and global states of distributed systems*. Univ., Department of Computer Science, 1988.
- [20] Minar Nelson. A survey of the ntp network, 1999.
- [21] Martin Olsson, Kjell Hansson, Ann-Marie Lundblad, and Marianne Cederblad. Sense of coherence : definition and explanation. *International Journal of Social Welfare*, 15(3) :219–229, 2006.
- [22] Su Ping. Delay measurement time synchronization for wireless sensor networks. *Intel Research Berkeley Lab*, 6 :1–10, 2003.
- [23] Michel Raynal and Mukesh Singhal. Logical time : Capturing causality in distributed systems. *Computer*, 29(2) :49–56, 1996.
- [24] J. Müller S. Bussmann. A negotiation framework for cooperating agents. *Proceedings of CKBS-SIG (CKBS’92)*, 1992.
- [25] Reinhard Schwarz and Friedemann Mattern. Detecting causal relationships in distributed computations : In search of the holy grail. *Distributed computing*, 7, 1994.
- [26] Philipp Sommer and Roger Wattenhofer. Gradient clock synchronization in wireless sensor networks. In *2009 International Conference on Information Processing in Sensor Networks*, pages 37–48. IEEE, 2009.
- [27] DJW Strümpfer, JF Gouws, and MR Viviers. Antonovsky’s sense of coherence scale related to negative and positive affectivity. *European Journal of Personality*, 12(6) :457–480, 1998.
- [28] Andrew Tanenbaum. *systèmes centralisés systèmes distribués*. 2ème édition Dunod, ISBN-10 : 210004554, 1998.
- [29] Paul Thagard. *Coherence in thought and action*. MIT press, 2000.
- [30] Paul Thagard. *Hot thought : Mechanisms and applications of emotional cognition*. MIT press, 2008.
- [31] Quang-Anh Nguyen Vu. *Cohérence et robustesse dans un système multiagent perturbé : application à un système décentralisé de collecte d’information distribué*. PhD thesis, Université Claude Bernard-Lyon I, 2012.