

L'approche Multi-Agents pour le contrôle adaptatif d'une constellation de satellites

Sidi-Mohammed SENOUCI¹, Dominique GAÏTI^{2,3}, Guy PUJOLLE³.

¹Laboratoire PRISM
Université de Versailles
45, Av. des Etats-Unis
78035 Versailles Cedex – France
ssm@prism.uvsq.fr

²LM2S
UTT
12, Rue Marie Curie
BP 2060 – 10010 Troyes cedex
Dominique.Gaiti@univ-troyes.fr

³Laboratoire LIP6
Université de Paris VI
8, rue du Capitaine Scott
75015 Paris – France
Guy.Pujolle@lip6.fr
Dominique.Gaiti@lip6.fr

Résumé

Un système LEO (*Low Earth Orbit*) est constitué d'une dizaine à une centaine de satellites défilant autour de la terre à une basse altitude et à une grande vitesse. Ces systèmes sont indépendants de la géographie de la terre et donc plus facile à remettre en œuvre en cas de désastre terrestre tel qu'un tremblement de terre. Cependant, ces systèmes posent des problèmes du fait de la mobilité des satellites et n'ont pas tout à fait les mêmes contraintes que dans le monde des radios mobiles, ce qui nécessite d'étudier et de valider de nouveaux mécanismes. Le développement d'architectures à base d'agents intelligents semble être une voie privilégiée pour arriver à maîtriser ces problèmes. L'objectif de ce travail est de concevoir et de réaliser un modèle multi-agents capable de contrôler cette complexité en exploitant les capacités d'interactions et de coopération des agents. L'implémentation de ce modèle a été faite en utilisant la plate-forme d'agents *MadKit*. Nous avons testé notre application sur une constellation constituée de 9 orbites contenant chacune 8 satellites répartis d'une façon équidistante à l'intérieur de l'orbite. Nous supposons que la constellation forme un système EFC (*Earth Fixed Cell*) que chaque satellite possède quatre liens inter satellitaires ISL (*Inter-Satellite Link*).

1 Introduction

Il est prévu que la composante satellite de la future norme UMTS (*Universal Mobile Telecommunications System*) va intégrer des constellations de satellites LEO/MEO pour servir une population d'utilisateurs demandant différentes qualités de service [1]. L'un des premiers objectifs de l'UMTS est non seulement d'offrir un service téléphonique, cause de l'échec de la première génération des constellations de satellites, mais aussi d'offrir le même ensemble de services que les réseaux fixes (ex. voix, données, vidéo, etc.).

L'intégration de ces nouveaux services engendrera une complexité du contrôle de ces systèmes. Le développement d'architectures à base d'agents intelligents semble être une voie privilégiée pour arriver à maîtriser cette complexité. Les constellations de satellites sont propices à ce type de contrôle par la distribution et l'éloignement géographique des différents éléments à contrôler. Les solutions de contrôle et de gestion de ces réseaux pourront passer par ce biais.

Notre objectif, dans ce travail, est de concevoir et de réaliser un environnement multi-agents capable de contrôler les paramètres d'une constellation de satellites. Nous proposons, donc, un modèle permettant de mettre en œuvre un contrôle adaptatif capable d'exploiter les capacités d'interactions et d'observations des agents pour contrôler la complexité (*handover, allocation de ressources, routage, garantie de QoS*). Nous décrivons l'architecture du modèle et nous spécifions l'interaction et la coopération entre les différentes entités. Ces agents interviennent sur les différents nœuds de la constellation.

Dans la section 2 nous introduisons la technologie agent en indiquant les origines du terme [8,10,12,13], puis en précisant les attributs que peut avoir un agent (autonomie, coopération, mobilité, réactivité, sécurité, etc.). Ensuite, nous présentons dans la section 3 les travaux de recherche, ainsi que les projets qui se fondent sur cette technologie agent pour le contrôle et la gestion des systèmes de télécommunications. Après une présentation de la problématique des constellations de satellites [1,2] dans la section 4, nous développons le modèle multi-agent proposé ainsi que son implémentation sous la plate-forme organisationnelle *MadKit* [25] dans la section 5. Et enfin, nous concluons et donnons les perspectives de ce travail dans la section.

2 Définitions relatives aux Agents

2.1 Origines

Les gens sont parfois perdus quand on évoque les agents intelligents car le terme est employé de différentes manières. Magedanz [3] donne une taxonomie du terme et identifie les quatre groupes suivants : agents locaux, agents réseaux, agents IAD et agents mobiles.

Cheikhrouhou [13], pour sa part, donne une explication à ces différentes définitions en donnant trois origines du terme : une origine qui vient du monde de l'Intelligence Artificielle (IA), une autre du monde Objets et une dernière du monde Réseaux.

1. *Monde Intelligence Artificielle* : les personnes travaillant en IA ont été influencées par les approches orientées objets en travaillant sur des entités logicielles réutilisables et plus flexibles.
2. *Monde Objets* : les personnes travaillant sur les objets ont voulu ajouter une certaine mobilité et autonomie à leurs objets, pour les appeler ensuite agents.
3. *Monde Réseaux* : Les personnes travaillant en réseaux utilisaient déjà le terme agent (agent SNMP et CMIP) ce qui n'a rien à voir avec le terme d'agent intelligent. La solution agent est prise en compte et est vue comme solution future par les compagnies telles que ORACLE pour leur architecture client/agent/serveur.

En fait, deux définitions du terme agent provenant de deux communautés distinctes, peuvent s'utiliser : celle de la communauté Génie Logiciel, qui définit un agent comme une entité autonome ou mobile pouvant migrer dans un réseau et accomplissant des tâches au nom de son propriétaire, et celle de la communauté Intelligence Artificielle qui, elle, considère des Systèmes Multi-Agents (SMA) composés de plusieurs entités distribuées capables de prendre des décisions basées sur la perception de leurs environnement et essayant d'atteindre un ou plusieurs buts. C'est cette dernière communauté et en particulier le domaine de l'Intelligence Artificielle Distribuée (IAD) qui a influencé en premier la technologie agent.

2.2 Architecture et Attributs des agents

Les systèmes multi-agents existants sont, en général, classés en trois principales familles : les systèmes cognitifs, les systèmes réactifs et les systèmes hybrides. Le comportement complexe des systèmes réactifs émerge de la coexistence et de la coopération des comportements simples. Chaque agent réactif dispose d'un protocole et d'un langage de communication.

Contrairement aux systèmes réactifs, les systèmes cognitifs se rapprochent le mieux du modèle de sociétés d'experts. Chaque agent cognitif a une représentation explicite ou implicite de croyances, d'intentions, d'actes de langages, de modèles des autres agents, etc.

Quant aux systèmes hybrides, ce sont des systèmes qui possèdent les deux caractéristiques des deux premier systèmes.

Parmi les attributs qui permettent de différencier ces agents et qui ont donné naissance à plusieurs types d'agents (agent autonome, agent collaborateur, agent mobile, etc.) on trouve :

- *Autonomie* : l'agent autonome décide lui-même quand et sous quelles conditions il accomplit une action. Les virus en sont des exemples ;
- *Communication* : c'est la possibilité de parler avec les autres et d'échanger des informations. Les agents peuvent communiquer directement, ou par l'intermédiaire d'un agent, ou en *broadcast* ou en utilisant un dépôt de données (*blackboard*) [19]. Parmi les langages les plus utilisés pour la communication inter agents, on trouve ceux basés sur la théorie des actes de langage [20]. Le langage KQML (*Knowledge Query and Manipulation Language*) est le plus utilisé dans la communauté des SMA ;
- *Collaboration/Coopération* : c'est la possibilité de travailler ensemble, communiquer et négocier ensemble pour résoudre un problème. Les agents peuvent coopérer suivant certains modèles : par partage de tâches, par commande, par appel d'offres et par compétition ;
- *Délibératif (cognitif)* : c'est un agent qui connaît les règles et les applique sans attendre des instructions. Il contient une représentation de l'environnement ;
- *Mobilité* : il existe deux niveaux de mobilité (i) l'*exécution distante (Remote Execution)* où l'agent est transféré au système distant où il sera activé et exécuté en entier et (ii) la *migration* où durant son exécution, l'agent se déplace de nœud en nœud pour accomplir une tâche progressivement ;
- *Apprentissage* : c'est la capacité de l'agent d'utiliser les nouvelles connaissances acquises par un ensemble d'expériences pour améliorer son comportement ;
- *Pro-Activité* : ces agents suivent un plan et exécutent des règles quand l'environnement atteint un seuil prédéfini (il y a une similarité avec les agents délibératifs) ;
- *Sécurité* : c'est la possibilité de différencier entre amis et ennemis ;
- *Planification* : l'agent organise par priorité les actions qu'il doit exécuter durant sa vie. Les agents ont un plan qui spécifie toutes les futures actions et interactions de chaque agent. Un plan peut être formalisé sous forme d'un automate qui décrit la conversation inter-agents [26] ;
- *Délégation* : l'agent demande à un autre agent d'exécuter une de ses tâches.

2.3 Les plates-formes d'agents

Les plates-formes d'agents sont des outils de construction et de développement d'agents. Dans [26], on trouve une classification de ces plates-formes en deux classes : celles qui supportent la mobilité (ex. *IBM Aglets*, *Grasshopper*, *Voyager*, *Odyssey*, etc.) et celles qui supportent les actes de langage (*JAFMAS*, *Swarm*, *MadKit*, etc.). Les auteurs donnent une comparaison entre ces différents outils. Dans [27], on retrouve une description de toutes les plates-formes commerciales et académiques existantes. Dans le paragraphe suivant, nous allons décrire la plate-forme *MadKit* utilisée dans notre travail.

2.3.1 La plate-forme *MadKit*

MadKit [17,25] est une plate-forme organisationnelle basée sur le modèle *Aalaadin* [17]. Ce modèle est basé sur trois concepts : agent, groupe et rôle. Il est organisé en groupes dynamiques, dont chacun est constitué d'agents qui communiquent entre eux pour pouvoir effectuer leurs tâches représentées par des rôles. L'aspect dynamique est assuré par le fait que l'agent peut changer de rôle dans le même groupe et voire quitter le groupe. Il peut aussi faire partie de plusieurs groupes.

Dans ce modèle, deux niveaux d'analyse sont définis :

- un niveau descriptif : dans ce niveau sont définis les agents, les groupes et les rôles possibles du système ;
- un niveau méthodologique : ce niveau définit les structures abstraites de l'organisation, ainsi que les différentes interactions entre les groupes.

3 Les agents dans les télécommunications

Dans cette section nous allons présenter les travaux de recherche ainsi que les projets qui se basent sur cette technologie agent pour le contrôle et la gestion des systèmes de télécommunications. *Hayzelden* [8] expose quelques travaux qui ont renforcé cette idée et les classe en quatre catégories et ceci suivant le type d'agent utilisé : approche Système Multi-Agent, approche Agent Mobile, approche économique et approche fourmi. Nous avons préféré ne considérer que deux catégories : l'approche Système Multi-Agents et l'approche Agent Mobile car elles englobent les deux autres.

Comme nous sommes plus intéressés par l'approche SMA, nous n'allons présenter dans la section suivante que des travaux et des projets qui lui sont relatifs. Les lecteurs intéressés par la deuxième approche 'Agent Mobile', peuvent se référer aux papiers [8,13].

3.1 Approche Système Multi-Agent (SMA)

3.1.1 Contrôle de congestion adaptatif de réseaux ATM

Dans [10,11], il est proposé un modèle permettant de mettre en œuvre un contrôle de trafic adaptatif dans les réseaux ATM. Ce modèle, sous forme de couches, comporte un niveau contrôle qui fait référence aux fonctions de contrôle que doit assurer un élément de communication, et un niveau méta-contrôle dont le rôle est d'observer l'état et le comportement de la couche contrôle ainsi que celui de l'environnement afin de prendre des décisions en adéquation avec l'évolution de l'état du réseau. Pour la mise en œuvre de la couche de contrôle, les auteurs proposent le mécanisme TRAC (*ThReshold Algorithm for Control*) dont le principe est de considérer deux niveaux de congestion qui permettent de déclencher différentes actions en fonction de l'occupation du tampon associé à l'élément de commutation. Pour mettre en œuvre la couche méta-contrôle, les auteurs proposent le mécanisme D-TRAC (*Dynamic-TRAC*) qui exploite les capacités d'interactions et d'observations des agents pour construire des croyances sur le comportement des agents du voisinage. Dans ce mécanisme, un ajustement des seuils de congestion est réalisé. En fonction de l'état du nœud, du type de message de notification reçu et du comportement des nœuds du voisinage, les seuils de congestion sont ajustés, ce qui permet l'exécution de l'action de contrôle la plus adéquate possible.

3.1.2 Une approche à base d'agent pour la gestion de la QoS dans Internet

Dans [6], les auteurs proposent une approche pour la négociation de la QoS dans Internet qui repose sur une nouvelle approche multi-agent. L'approche standard implique que le manager doit avoir connaissance des différents fournisseurs de services, des composants du système et de toutes les ressources dans son environnement. Mais vu que l'Internet est un environnement hétérogène avec des millions d'ordinateurs, cette approche peut avoir plusieurs limitations.

Les auteurs définissent quatre types d'agents : agent *User*, agent *Service*, agent *Broker* et agent *Facilator* et qui communiquent en utilisant KQML. Les clients (agents *User*) et les fournisseurs de services (agents *Service*) s'engagent dans une communication par l'intermédiaire d'agents *Broker* et d'agents *Facilator*. Ces

derniers obtiennent les informations sur la capacité des fournisseurs de service et agissent comme les seuls points pour fournir ces informations aux clients.

3.1.3 Les agents intelligents pour le contrôle des systèmes multimédias

Dans [5], les auteurs utilisent une approche agent intelligent pour satisfaire la QoS demandée par les applications multimédias, et proposent une gestion de la QoS sur tous les éléments de la liaison de bout en bout en utilisant des agents. L'intelligence de ces agents consiste en la possibilité d'adaptation et de coopération. Les auteurs utilisent trois types d'agents : un agent *QoS*, un agent *Host* et un agent *Network*. Quand une violation de la QoS apparaît dans un certain composant, l'agent QoS associé essaye de trouver une solution locale en modifiant des paramètres et en faisant une re-allocation des ressources. Si c'est impossible, il dialogue avec d'autres agents *QoS* de la même connexion pour demander d'autres ressources et pour compenser la dégradation courante. Si le système n'a plus de ressources suffisantes, une renégociation est initialisée avec les applications.

3.1.4 La gestion coopérative de la QoS pour les applications multimédias

Dans [14], les auteurs proposent une approche coopérative de gestion de la QoS pour une application multimédia. Il prennent comme exemple d'applications, l'enseignement à distance. L'idée est d'installer des agents dans chaque nœud du réseau et dans chaque système d'extrémité participant à l'application. Les auteurs utilisent trois types d'agents : un agent *receveur*, un agent *émetteur*, et un agent *QoS*. Ces agents peuvent communiquer entre eux, et donc localiser les problèmes liés à la QoS et négocier pour une utilisation plus optimale des ressources.

Les auteurs décrivent ce nouveau mécanisme de gestion en terme de trois fonctions : négociation, adaptation et renégociation :

- Négociation : la négociation apparaît quand un nouvel utilisateur rentre dans l'arbre du *multicast*. L'agent local contacte un agent dans chaque groupe *multicast* pour avoir toutes les informations sur les flux disponibles dans cette partie de l'arbre et les offre à l'utilisateur qui va sélectionner les flux qu'il désire et les remet à cet agent local. Ce dernier a l'information sur la manière de délivrer le flux à l'utilisateur ;
- Adaptation : l'adaptation apparaît quand la QoS est violée et elle consiste en la destruction d'une partie seulement de l'ancien arbre *multicast* et la construction d'un nouveau ;
- Renégociation : l'utilisateur initie une renégociation pour redemander une bonne qualité ou pour réduire le coût de la session courante ;

3.1.5 Gestion intelligente du routage dans les réseaux ATM

Dans [9], les auteurs font une analogie entre le système immunitaire humain et le système de gestion de réseau ATM. Le problème consiste en l'adaptation du système aux événements qui apparaissent régulièrement. Le système Multi-Agents contient des agents statiques appelés «*connection routing agents*» localisés dans chaque nœud du réseau. Ils ont une table contenant toutes les routes disponibles entre la source et la destination. Ils ont aussi une table «*Immunity Table*» contenant les séquences de problèmes apparus auparavant et les remèdes associés. Quand une demande de connexion est refusée, l'agent essaye de satisfaire cette demande en trouvant une nouvelle route. Il génère après une séquence d'exception contenant la date de l'événement, le type de trafic, la bande passante demandée, la nouvelle route choisie, etc. Cette séquence est enregistrée dans la table puis diffusée aux différents nœuds d'accès dans le réseau (*clonage*). Si l'événement ré-apparaît plusieurs fois, la séquence devient en début de la table et donc la première à retrouver en cas de problème de connexion.

4 Les constellations de satellites

Face au succès croissant des services multimédias et ceux des réseaux mobiles, les opérateurs de télécommunication sont vivement intéressés à la perspective d'offrir de tels services à leurs abonnés fixes ou mobiles par le biais des réseaux de satellites LEO ou MEO (basse ou moyenne orbite). Un système LEO est constitué d'une dizaine à une centaine de satellites tournant autour de la terre à une basse altitude. Ces systèmes sont indépendants de la géographie de la terre et sont plus facile à remettre en œuvre en cas de désastre terrestre tel qu'un tremblement de terre [2].

Il existe de nombreux projets de réseaux satellitaires (environ une trentaine). Parmi ces projets industriels, on peut noter les suivants : le projet *Iridium* [22], lancé par *Motorola* en 1990, comportant 66 satellites qui a pour objectif de ne fournir que des services téléphoniques. Vu que ce projet a connu l'échec, *Motorola* a

annoncé le lancement du projet Iridium 2 capable d'offrir des services multimédias. Le système *Skybridge* [23], lancé par *Alcatel Telecom*, est constitué de 64 satellites et offre des services multimédias. Le système *Teledesic* [24], appuyé par *Bill Gates* et *Boeing*, fournira des services d'accès à Internet par le biais de ses 288 satellites dès 2002. Il offrira des débits pouvant aller jusqu'à 2 Mbit/s et prendra en charge des flux multimédias.

Cependant, les systèmes LEO posent des problèmes particuliers du fait de la mobilité des satellites et de leur faible altitude. Ainsi, le problème du basculement des satellites n'a pas les mêmes contraintes que celles du monde des radios mobiles, ce qui nécessite d'étudier et de valider de nouveaux mécanismes. D'autre part, il est nécessaire de garantir, pour chaque connexion, une qualité de service négociée dans le contrat de service. Les mécanismes de basculement de satellite doivent préserver cette qualité de service. La qualité de service à maintenir impose des contraintes plus ou moins fortes selon le type de connexion en tenant compte du fait que le réseau doit être capable de supporter du trafic multimédia. Pour assurer le succès du marché des liaisons par satellites défilants, l'optimisation des mécanismes, en terme de simplicité mais aussi de sûreté de fonctionnement, est une obligation pour aboutir à un service de qualité.

Excessivement, peu de développements ont été effectués sur les plates-formes intelligentes de contrôle. Les constellations de satellites sont propices à ce type de contrôle par la distribution et l'éloignement géographique des différents éléments à contrôler. Ce travail permet d'expérimenter cette solution en grandeur réelle et dans un contexte favorable.

5 Implémentation

Nous avons testé notre application sur une constellation constituée de 9 orbites contenant chacune 8 satellites répartis de façon équidistante à l'intérieur de l'orbite. Chaque satellite possède quatre liens inter satellitaires (ISL) et donc, quatre voisins, deux sur la même orbite (appelés successeur et prédécesseur) et les deux autres sur les plans adjacents se trouvant à l'est et à l'ouest (appelés droit et gauche). La répartition initiale des satellites au sein de leur orbite est donnée sur la *figure 1*. Nous supposons que la constellation forme un système EFC (*Earth Fixed Cell*) ce qui signifie que les empreintes (cellules de couverture) qu'elle forme sur la surface de la terre sont fixes au sol.

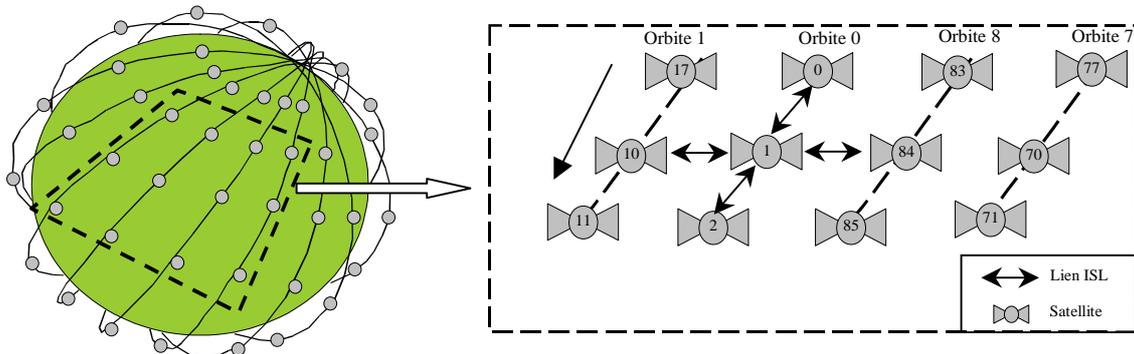


Fig. 1. La répartition initiale des satellites

Ce travail est implémenté sur la plate-forme d'agent *MadKit* décrite dans la section 2. Nous avons opté pour cette plate-forme qui répond exactement à nos besoins de structuration et d'organisation dynamique de la constellation. L'aspect dynamique de la constellation est pris en charge par la dynamique des agents au sein de leurs groupes. N'ayant aucune restriction sur la nature des agents, *MadKit* nous permet de définir en toute liberté nos agents.

Dans cette section, nous allons présenter le modèle que nous avons proposé pour le contrôle d'une constellation de satellites. Nous allons donner, en premier, l'organisation de nos agents et le rôle de chacun au sein des différents groupes. Nous donnons, ensuite, un exemple de déroulement d'une communication entre deux utilisateurs incluant un *handover* inter-satellitaire pour démontrer la dynamique des groupes lors d'une communication.

5.1 Niveau descriptif

5.1.1 Les agents

Nous avons utilisé trois types d'agents :

- les agents *satellites* qui représentent les satellites de la constellation. Les paramètres qui caractérisent un satellite sont : l'orbite à laquelle il appartient, sa position à l'intérieur de l'orbite et l'empreinte terrestre

(zone) qu'il couvre. Nous rappelons que les empreintes sont fixes. Pour notre simulation, nous avons supposé qu'une empreinte est couverte par un seul satellite. La notation satellite ij définit le satellite qui, au démarrage de l'application, avait la position j à l'intérieur de l'orbite i ;

- les agents *utilisateurs* qui représentent les utilisateurs qui sont comme nous l'avons déjà précisé fixes ;
- les agents *empreintes* qui représentent les empreintes de la constellation. Leur nombre est égal à celui des satellites. Chaque empreinte ij est couverte au démarrage de l'application par le satellite ij .

5.1.2 Les groupes et les rôles

Ce sont les groupes qui définissent l'organisation des différentes entités de l'application de manière logique et qui leurs donnent la possibilité de communiquer. Un utilisateur, par exemple, n'a pas besoin de connaître tous les utilisateurs qui existent, il lui suffit de connaître son interlocuteur.

Nous avons défini quatre types de groupes :

- le groupe voisinage d'un satellite 'i' : comme nous avons considéré que la constellation comprend quatre liens inter-satellites, le groupe voisinage regroupe le satellite i avec ses voisins. Il y a autant de groupes voisinage qu'il y a de satellites. Un satellite 'i' joue le rôle de créateur dans le groupe "voisinage de i" mais dans les groupes voisinage de ses voisins, il joue les rôles suivants : successeur, prédécesseur, droit et gauche selon sa position (cf. figure 2) ;

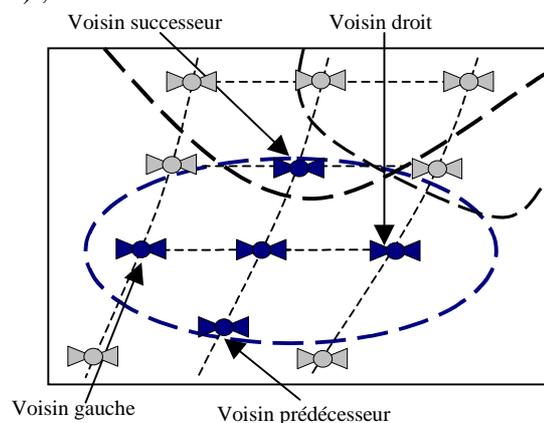


Fig. 2. Les groupes voisinage

- le groupe communication (satellite1, satellite2, utilisateur1, utilisateur2, satellites intermédiaires) regroupe les différentes parties participant à une communication donnée. L'utilisateur source (utilisateur1) et l'utilisateur destination (utilisateur2), le satellite source (satellite1) et le satellite destination (satellite2) et les différents satellites intermédiaires. Le nombre de groupes communication est quelconque ;
- le groupe couverture (zone i , satellite j) regroupe l'empreinte terrestre i avec le satellite j qui la couvre ;
- le groupe appartenance (utilisateur i , zone j) comprend l'utilisateur et l'empreinte terrestre à laquelle il appartient.

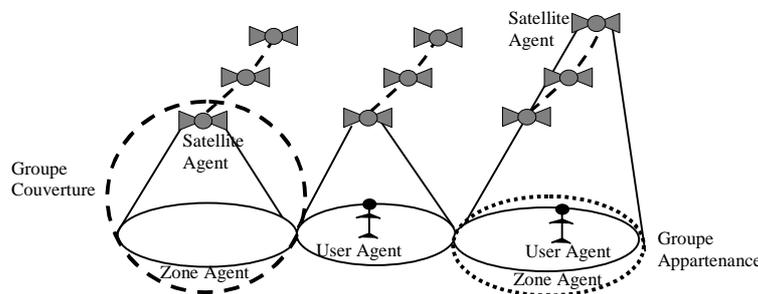


Fig. 3. Les groupes couverture et appartenance

5.2 Niveau méthodologique

5.2.1 Déroulement d'une session

Le déroulement d'une session est réalisé suivant cinq étapes :

- L'application démarre par le lancement d'un certain nombre d'agents satellites, regroupés selon leur voisinage ;

- b) Les agents utilisateurs sont lancés à des intervalles de temps égal à t_1 ;
- c) Début de la communication et calcul de la route entre la source et la destination. On trouve différents algorithmes de routage et ils se distinguent suivant leur degré d'adaptation aux spécificités des constellations de satellites [4]. Pour ce travail, nous avons utilisé l'algorithme MHA (*Minimum Hops Algorithm*) [4] qui trouve le chemin avec un nombre de sauts d'orbites minimum et qui peut être implémenté simplement en utilisant l'algorithme du plus court chemin de *Dijkstra* ;
- d) Prise en compte de la dynamique des satellites et donc de l'apparition possible d'un *handover* ;
- e) Fin de la communication.

Dans la section qui suit, nous décrivons un exemple de déroulement d'une session entre deux utilisateurs en prenant en compte le cas d'apparition de *handover*.

5.2.2 Exemple de gestion d'une communication

Nous allons illustrer dans ce qui suit le déroulement d'une communication entre un utilisateur se trouvant dans l'empreinte n°11 et un deuxième utilisateur se trouvant dans l'empreinte n°70 (cf. figure 1).

La communication démarre au temps t et a pour durée $t+2 T_p$ (T_p est le temps nécessaire pour l'apparition d'un *handover*, et, puisque nous avons considéré un système EFC, c'est le temps pour que le satellite puisse avoir la position dans laquelle se trouvait son prédécesseur).

a) A l'instant t (début de la communication)

L'utilisateur 11 entre en communication avec le satellite qui couvre l'empreinte terrestre n°11. Il crée les deux groupes appartenance (utilisateur 11, zone 11) et couverture (zone 11, satellite 11). Si le satellite a la possibilité de prendre en charge la communication en question, il répond par un acquittement positif qu'il envoie à l'utilisateur 11 par l'intermédiaire de l'empreinte 11.

L'utilisateur 11 crée l'agent utilisateur 70 qui, à son tour, essaie d'entrer en liaison avec le satellite qui couvre la zone terrestre 70. Ce dernier crée, à son tour, les deux groupes appartenance (utilisateur 70, zone 70) et couverture (zone 70, satellite 70).

Le satellite source (11) déroule l'algorithme de routage MHA, décrit ci-dessus, afin de calculer le chemin que va suivre la communication. Ce procédé est effectué en deux étapes. Une première étape consiste à faire des sauts inter-orbites jusqu'à ce que l'orbite du satellite destination soit atteinte. Une deuxième étape consiste à faire des sauts intra-orbites pour arriver au satellite destination.

Pour les sauts inter-orbites, le satellite 11 doit choisir entre les deux séquences d'orbites 1,0,8,7 et 1,2,3,4,5,6,7. Cette dernière exploite les voisins gauches alors que la première exploite les voisins droits. La première séquence est choisie car elle met en jeu moins d'orbites. Le satellite 11 envoie donc un message à son voisin gauche, en l'occurrence le satellite 2. Ce dernier demande d'abord à faire partie du groupe communication (satellite 11, satellite 70, utilisateur 11, utilisateur 70, null). Il joue le rôle de satellite intermédiaire, lance ensuite une requête auprès du satellite 85 pour qu'il fasse à son tour partie de la communication.

Lorsque le satellite 71 reçoit le message du satellite 85, il s'aperçoit qu'il appartient à l'orbite du satellite destination (70). L'algorithme de routage calcule le chemin à l'intérieur de cette orbite en minimisant le nombre de sauts intra-orbites. Il doit choisir entre les deux séquences suivantes de satellites : 70, 71 et 70,77,76,75,74,73,72,71. La première est évidemment celle qui est choisie et le groupe communication (satellite 11, satellite 70, utilisateur 11, utilisateur 70, (satellite 2, satellite 85, satellite 71)) est maintenant complet (cf. figure 4).

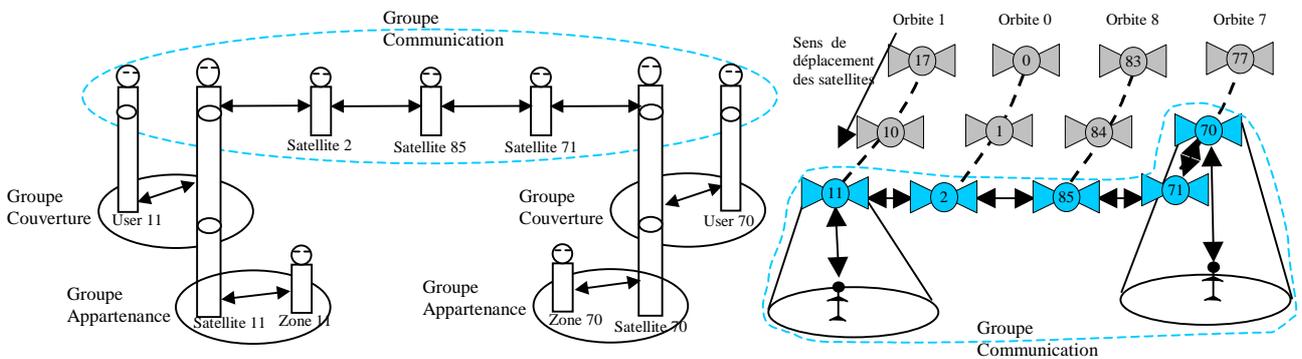


Fig. 4. Le groupe communication au temps t

b) A l'instant $t + T_p$ (premier handover)

Le problème qui se pose est que l'utilisateur 11 n'est plus couvert par le satellite 11 mais par son successeur, le satellite (10), et ce scénario est pareil pour le satellite 70 couvert à présent par le satellite 77. Les groupes couverture doivent donc être mis à jour, ce qui nécessite de recalculer le chemin de la communication. Ce dernier sera constitué des successeurs des satellites qui composaient le chemin initial. Le satellite 11 sera remplacé par exemple par le satellite 10 qui va prendre en charge toutes les communications dont s'occupait le satellite 11. Le chemin est donc constitué des satellites suivants 10,1,84,70,77.

Le groupe communication (satellite 11, satellite 70, utilisateur 11, utilisateur 70, (satellite 2, satellite 85, satellite 71)) disparaît et est remplacé par le groupe communication (satellite 10, satellite 77, utilisateur 11, utilisateur 70, (satellite 1, satellite 84, satellite 70)) (cf. figure 5).

c) A l'instant $t + 2 T_p$ (fin de la communication)

Le groupe communication (satellite 10, satellite 77, utilisateur 11, utilisateur 70, (satellite 1, satellite 84, satellite 70)) disparaît en mettant ainsi fin à la communication en question et les satellites seront remplacés par leurs successeurs. Les groupes appartenance (utilisateur 11, zone 11) et (utilisateur 70, zone 70) disparaissent aussi.

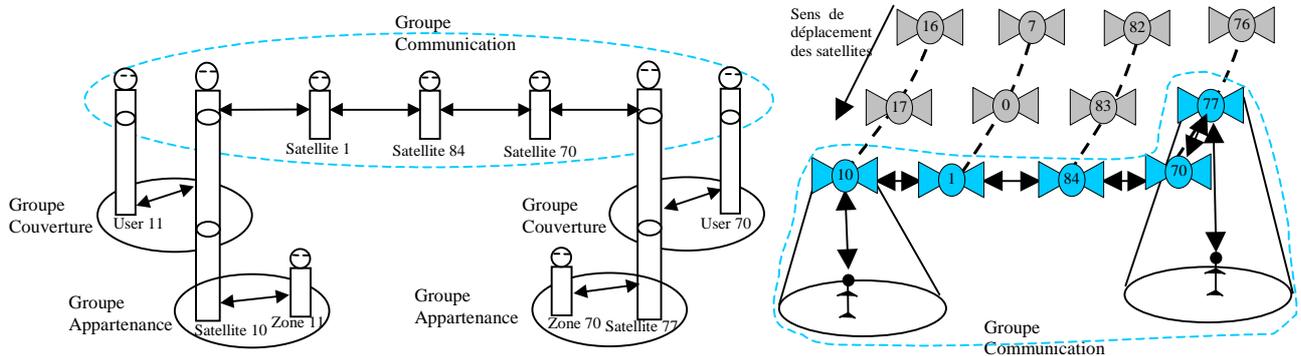


Fig. 5. Le groupe communication au temps $t + T_p$

6 Conclusion et perspectives

Nous avons présenté, dans ce papier, une modélisation multi-agent d'une constellation de satellites à basse orbite LEO, en utilisant la plate-forme organisationnelle *MadKit*. Ce modèle a permis de mettre en œuvre un contrôle adaptatif capable d'exploiter les capacités d'interaction et d'observation des agents, intervenant aux différents nœuds, pour contrôler la complexité de la constellation. Ce travail a permis de voir la dynamique de l'organisation de la constellation et la complexité de gérer et de router les informations (son et/ou image) de bout en bout. L'aspect dynamique de la constellation (mobilité des satellites, *handover*) est prise en charge par le fait que l'agent peut changer de rôle dans le même groupe et quitter le groupe.

Nous avons décrit l'architecture du modèle SMA et spécifié l'interaction entre les agents car, dans ce premier travail, nous n'avons considéré que la partie réactive de nos agents hybrides. Notons que le comportement intelligent des agents réactifs émerge de l'interaction des comportements simples de chaque agent. Dans nos travaux futurs, nous nous intéresserons à l'architecture de l'agent lui-même et donc à la partie délibérative de nos agents. Ainsi, on compte associer à ces agents des fonctions d'observation, de raisonnement et d'apprentissage [18] pour réaliser des agents plus adaptatifs et, donc, installer un méta-contrôle au-dessus des différents algorithmes de contrôle (*handover*, allocation de ressources, etc.). Nous avons, donc, défini un squelette du système de contrôle dont l'enrichissement va donner un modèle complet de contrôle de la constellation.

Références

- [1] Gérard Maral, Joaquin Restrepo, Enrico Del Re, Romano Fantacci, Giovanni Giambene, 'Performance Analysis for a Guaranteed Handover Service in an LEO Constellation with a SFC System', IEEE Transactions on Vehicular Technology, Vol. 47, No. 4, p. 1200-1214, Nov. 1998.
- [2] Bezalel Gavish, 'LEO/MEO systems – Global mobile communications systems', Telecommunication Systems 8, p. 99-141, 1997.
- [3] T. Magedanz, K. Rothermel, S. Krause, 'Intelligent Agents : An Emerging Technology for Next Generation Telecommunications', IEEE Infocom 96, p. 464-472, San Francisco USA, Mars 1996.

- [4] John C. S. Lui, Peter Tsz-Shing Tam, Hak-Wai Chan, 'Routing and Channel Reservation Strategies for a LEO Satellite System', IEEE Infocom 2000, Tel-Aviv Israel, Mars 2000.
- [5] N. Agoulmine, F. Nait-Abdesselam, 'Intelligent Agent Architecture as A Flexible Solution for Multimedia System and Network Control', Mini-Conférence Multimedia Management, Networking 2000, Paris France, Mai 2000.
- [6] T. K. Mamadou, Tatsuo Nakajima, 'An agent-based framework for large scale Internet applications', Networking 2000, Lecture Notes in computer science Vol. 1815, p. 632-642, Paris France, Mai 2000.
- [7] John Bigham, Laurie Cuthbert, Alex Hayzelden et Zhiyuan Luo, 'Multi-Agent System for Network Resource Management', 6th International Conference on Intelligence and Services in Networks, IS&N'99. Lecture Notes in Computer Science, Vol. 1597, p. 514-526, Avril 1999.
- [8] Alex Hayzelden et John Bigham, 'Agent Technology in Communications Systems : An Overview', Revue Knowledge Engineering, Vol. 14, No. 4, 1999.
- [9] Alex Hayzelden et John Bigham, 'Intelligent Route Management in ATM Networks using Distributed Software Agents', Proceedings of Fifteenth UK Teletraffic Symposium on Performance Engineering in Information Systems, Mars 1998.
- [10] N. Boukhatem, 'L'approche multi-agents pour un contrôle de congestion adaptatif de réseaux ATM', Thèse de doctorat, Université de Versailles, 1998.
- [11] D. Gaiti, N. Boukhatem 'Cooperative Congestion Control Schemes in ATM Networks', IEEE Communications Magazine, Vol. 34. No. 11, p.102-110, 1996.
- [12] M. Wooldridge et N.R. Jennings, 'Intelligent Agents : Theory and Practice', Revue Knowledge Engineering, Vol. 10, No. 2, p. 115-152, 1995.
- [13] M. M. Cheikhrouhou, Pierre Conti, Jacques Labetoulle, 'Intelligent Agents in Network Management, a State-of-the-Art', Networking and Information Systems 1, p. 9-38, 1998.
- [14] Stefan Fischer, Abdelhakim Hafid, Gregor von Bochmann and Hermann de Meer, 'Cooperative QoS Management for Multimedia Applications', ICMCS, p. 303-310, Juin 1997.
- [15] M.A. Gibney, N.R. Jennings, 'Dynamic Ressource Allocation by Market-Based Routing in Telecommunications Networks', Intelligent Agents for Telecommunications Applications IATA, p. 102-117, Paris France, 1998.
- [16] S. Appleby, S. Steward, 'Mobile Software Agents for Control in telecommunications networks', BT Technology Journal, Vol. 12 No. 2, p. 104-113, Avril 1994.
- [17] O. Gutknecht, J. Ferber, 'The MadKit agent platform architecture' , 1st Workshop on Infrastructure for Scalable Multi-Agent Systems , Barcelone, Juin 2000.
- [18] Junhong Nie, Simon Haykin, 'A Q-Learning-Based Dynamic Channel Assignment Technique for Mobile Communication Systems', IEEE Transactions on Vehicular Technology, Vol. 48, No. 5, p. 1676-1687, Sept. 1999.
- [19] M.R. Genesereth et S.P. Ketchpel, 'Software Agents', Communications of the ACM 37 (7), p. 48-53, 1994.
- [20] J.L. Austin (ed.), '*How to do Things With Words*', Oxford University Press, 1962.
- [21] S. Appleby et S. Steward, 'Software Agents for Control', in P. Cochrane and P. Mealthy, Eds., *Modeling Future Telecommunication Systems*. Chapman & Hall, 1994.
- [22] Iridium website <http://www.iridium.com>
- [23] Skybridge website <http://www.skybridgesatellite.com>
- [24] Teledesic website <http://www.teledesic.com/overview/frover.html>
- [25] MadKit website <http://www.madkit.org>
- [26] JAFMAS website <http://www.eecs.uc.edu/~abaker/JAFMAS>
- [27] AgentBuilder website <http://www.agentbuilder.com>