



HAL
open science

Un protocole de session pour réseaux de capteurs sans fil

Said Harchi, Thierry Divoux, Jean-Philippe Georges

► **To cite this version:**

Said Harchi, Thierry Divoux, Jean-Philippe Georges. Un protocole de session pour réseaux de capteurs sans fil. 10ème Conférence Francophone de Modélisation, Optimisation et Simulation, MOSIM'14, CRAN, Nov 2014, Nancy, France. hal-01071430

HAL Id: hal-01071430

<https://hal.science/hal-01071430v1>

Submitted on 14 Oct 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UN PROTOCOLE DE SESSION POUR RESEAUX DE CAPTEURS SANS FIL

Saïd HARCHI^{1,2}, Thierry DIVOUX^{1,2}, Jean-Philippe GEORGES^{1,2}

¹ Université de Lorraine, CRAN, UMR 7039, Campus Sciences,
BP 70239, Vandoeuvre lès Nancy Cedex, 54506, France.

² CNRS, CRAN, UMR 7039, France.

said.harchi@univ-lorraine.fr, thierry.divoux@univ-lorraine.fr, jean-philippe.georges@univ-lorraine.fr

RESUME : *Les réseaux de capteurs sans fils trouvent des applications dans la surveillance de grands systèmes (ponts, tunnels, ...) ou dans la gestion de catastrophes (feux de forêt, marées noires, ...). Dans certaines de ces applications, la topologie du réseau est dynamique : soit les capteurs sont géographiquement mobiles (dispersion d'une nappe de pétrole), soit les conditions environnementales évoluent et modifient les capacités de communication des capteurs.*

Aussi, d'un système connexe, on peut évoluer vers un système partitionné (clustérisé) qui présente une rupture de la connectivité globale, et donc du système d'information. Une solution consiste à utiliser un (ou des) collecteur(s) des mesures (par exemple un robot mobile) qui va régulièrement visiter les clusters disjoints afin de rétablir une connectivité discrète pour reconstituer un système d'information cohérent. Nous proposons un protocole de couche session permettant au collecteur de reconstruire le contexte de communication avec les clusters précédemment visités, sachant qu'ils ont pu évoluer en position, fusionner ou se scinder. Pour le valider, nous avons défini un modèle de nœud capteur que nous avons intégré à l'environnement de simulation Riverbed Opnet Modeler.

MOTS-CLES : *Réseaux de capteurs sans fil, protocole de session, Riverbed Opnet Modeler.*

1 INTRODUCTION

Dans cet article, pour illustrer notre propos, nous choisissons une application d'observation d'une marée noire. Dès 1972, le programme mondial IGOSS de surveillance continue de la pollution des mers a été initié (*Integrated Global Ocean Services System*). En 1979, un rapport suggère de déployer des capteurs sur les nappes de pétrole pour étudier l'évolution des pollutions par hydrocarbures (Levy E. M., 1979). L'émergence plus récente des réseaux de capteurs sans fil conforte cette idée. En effet, par rapport au survol aérien visuel classique, ils vont permettre un suivi continu et temps réel de la dérive des nappes, de leur taille, de leur consistance, etc.

Dès le début de la marée noire, on disperse donc aléatoirement de nombreux nœuds capteurs pour l'observer. La difficulté majeure à relever est alors de faire face à la dynamique du système faisant varier la topologie du réseau. Elle est principalement causée par deux types de mobilité qui sont :

- la mobilité des nappes qui provoque un changement topologique du réseau pouvant créer une rupture de communication. En effet, une fois déversé, le pétrole a immédiatement tendance à s'étaler à la surface de l'eau et former des nappes, qui en théorie, forment un cercle régulier dont le diamètre augmente huit fois dans l'heure qui suit le déversement, puis cinq fois avant la fin de la semaine. Ensuite, le vent, les vagues et la force de Coriolis liée à la rotation de la Terre contribuent à l'étirement, au déplacement et à la fragmentation de la nappe jusqu'à sa décomposition en gouttelettes ;

- la mobilité des nœuds eux-mêmes déployés sur les nappes de pétrole. En effet, sous l'effet de facteurs externes comme le vent, les nœuds peuvent se déplacer de façon aléatoire ce qui peut affecter les liaisons radio lorsque les nœuds sont éloignés les uns des autres. De même, l'effet de facteurs endogènes comme l'épuisement des batteries ou la défaillance des nœuds peut influencer la topologie du réseau et causer des problèmes d'acheminement des informations.

La dynamique du réseau peut donc conduire à une perte de connectivité partielle ou totale pour un ou plusieurs nœuds conduisant à une incohérence du système d'information global (graphe de communication non connexe). Nous proposons donc une architecture en 3 niveaux pour un système d'information cohérent de mesure, de collecte, et de traitement des données relatives à la propagation de nappes de pétrole en mer (figure 1) :

- la mesure (viscosité, température, position, ...) est réalisée par chaque capteur, et transmise à un chef de cluster élu pour représenter un certain nombre de nœuds, afin de hiérarchiser les communications pour en réduire le nombre et les localiser. En effet, on pourra être amené à déployer plusieurs dizaines de milliers de capteurs, et le système peut rapidement atteindre des dimensions importantes. Le processus électif et la métrique de partitionnement retenue sont rapidement exposés dans le paragraphe 2, car nécessaires à la compréhension, sans être toutefois l'objet de l'article ;

- la collecte est réalisée par un (ou plusieurs) collecteur(s) (robot, avion, ...) qui va visiter à une période fixe ou variable les chefs de cluster pour recueillir les mesures. Comme les clusters sont mouvants, en position, en taille, en nombre, et en représentation (le chef peut avoir changé) et que nous ne pouvons pas avoir un continuum du suivi de l'évolution des nappes, nous proposons de mettre en œuvre une couche abstraite dite de « session » dont l'objectif est de limiter la trajectoire du collecteur aux chefs des différents clusters, et de recontextualiser chacun de leurs retours. La trajectoire du collecteur sera définie en fonction de paramètres de session spécifiques. Le paragraphe 3 présentera la formalisation d'une session. Le paragraphe 4 détaillera les scénarios et résultats de simulation. La discussion menée dans le paragraphe 5 donnera des indications pour ajuster les paramètres de session ;

- le traitement est réalisé à un niveau central dit « système d'information » (SI), approvisionné en données par le(s) collecteur(s). Dans cet article, nous n'étudions pas cet aspect du problème global. Il ne sera donc pas détaillé davantage.

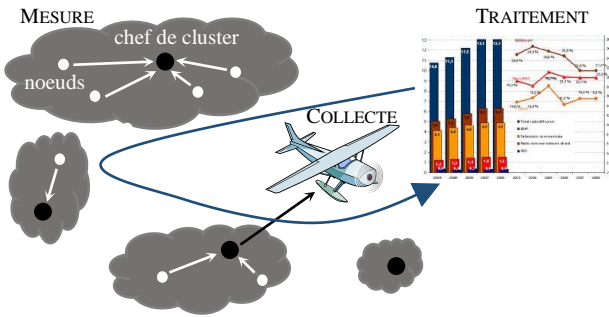


Figure 1 : L'architecture à 3 niveaux

2 LE PARTITIONNEMENT

Les caractéristiques des réseaux de capteurs nécessitent la mise en place de mécanismes d'auto-organisation consistant à partitionner le réseau en un certain nombre de groupes de nœuds homogènes appelés « clusters ». En effet, le nombre de nœuds, leur faible capacité de transmission et de traitement, leur mobilité, leur énergie limitée, ..., ne permettent pas à chacun de communiquer directement avec le « puits » où doivent converger toutes les données. Un cluster est un sous-ensemble de nœuds connexes, et le clustering est le processus de regroupement des nœuds en clusters disjoints (figure 2).

Un nœud particulier appelé chef de cluster ou *Cluster Head* (CH), permet de coordonner les membres de son cluster, d'agréger et/ou de traiter les données mesurées et de les transmettre au collecteur de données. Le chef de cluster est préalablement choisi, ou élu par tous les nœuds du réseau selon une métrique bien particulière ou une combinaison de métriques.

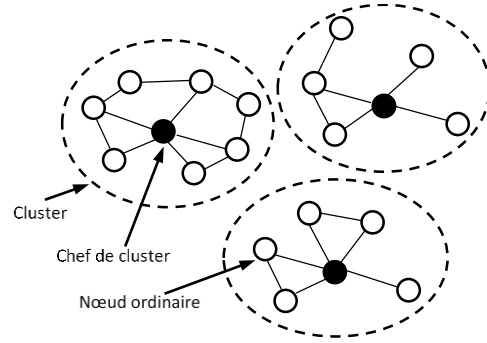


Figure 2 : Partitionnement en clusters

Dans (Harchi et al., 2012) après avoir étudié et analysé les nombreuses métriques publiées dans la littérature (Abbasi A.A. et Younis M., 2007), (Arboleda L.M.C. et Nasser N., 2006), (Kumarawadu P., et al., 2008), nous avons proposé un algorithme d'élection du chef de cluster basé sur une métrique adaptée aux caractéristiques spécifiques de notre application. Celle-ci, à minimiser, intègre les paramètres énergie résiduelle, densité et mobilité des nœuds :

$$Poids_u(t) = (1 - \alpha)M_u(t)/(\varepsilon_u(t) \cdot \alpha \rho_{u,1}(t))$$

avec

$\varepsilon_u(t)$: l'énergie résiduelle du nœud u.

α : un paramètre qui dépend de l'application.

$$\rho_{u,1}(t) = \frac{N_{u,1}(t-\Delta t)}{N_{u,2}(t-\Delta t)}$$
 densité du nœud u (Mittou, 2006)

$N_{u,k}(t)$: nombre de voisins à k-sauts du nœud u

$$M_u(t) = \frac{(|d_u(t) - d_u(t-\Delta t)|)}{\Delta t}$$
 : mobilité du nœud u

avec : $d_u(t) = \frac{1}{|N_{u,1}(t)|} \sum_{v \in N_{u,1}(t)} dist_{(u,v)}(t)$, et :

$$dist_{(u,v)}(t) = \sqrt{(X_u(t) - X_v(t))^2 + (Y_u(t) - Y_v(t))^2}$$

Notre application à forte dynamique suggère que nous choissions d'élire le chef dans un k-voisinage le plus petit possible. La métrique étant basée sur une 1-densité, les membres du cluster seront au maximum à 2 sauts du chef.

L'algorithme déroulé par chaque nœud, est illustré par l'automate à état finis de la figure 3. Il met en œuvre des messages *Hello* pour que chaque nœud connaisse son voisinage. Ces messages l'informent des paramètres densité, mobilité, et énergie de chaque voisin, lui permettant de calculer sa métrique $m(i)$ afin de la comparer à celle des voisins $m(hello_j)$. Des messages *Invite* et *Adhésion* permettent respectivement au chef de cluster et à un nœud ordinaire, de proposer de rejoindre le cluster et d'accepter cette proposition.

Nous avons validé l'algorithme en l'insérant dans un modèle de nœud sous l'environnement de simulation réseau Riverbed (Opnet) Modeler. Nous avons également développé un outil « photographe » (Figure 4) nous permettant de visualiser la dynamique de formation des clusters.

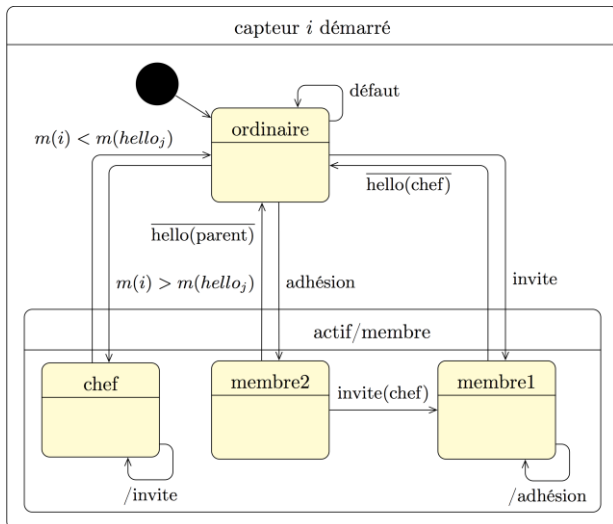


Figure 3 : Diagramme d'états d'un nœud

Nous avons ainsi pu montrer comment choisir en fonction des spécificités applicatives (vitesse des nœuds, ...) le paramètre de pondération densité/mobilité α qui assure une bonne stabilité du partitionnement : maximisation du temps pendant lequel un nœud reste chef de cluster, tout en minimisant le nombre de clusters formés. Cette recherche de stabilité permettra également d'augmenter la durée de vie opérationnelle des capteurs.

3 JUSTIFICATION ET FORMALISATION DU PROTOCOLE DE SESSION

Le niveau « mesure » étant maintenant auto-organisé, nous pouvons étudier le niveau « collecte » objet de cet article. Ce niveau est constitué d'éléments mobiles ayant pour tâche de relever les informations du niveau « mesure » et les transmettre au niveau « traitement ». Mais la dynamique de l'application a pour conséquence qu'entre deux passages successifs dans une même région, ils vont trouver une situation qui aura potentiellement énormément évolué. Ce changement de contexte est un frein à la restauration d'un système d'information cohérent qui doit s'affranchir des ruptures de connectivité du niveau « mesure ». Rappelons que le collecteur a pour mission de recouvrer une connectivité discrète lorsque la dispersion des nœuds rend le graphe de communication global non connexe. Nous illustrons cette problématique à travers l'exemple suivant :

Considérons deux clusters C1 et C2 et CH(C1) et CH(C2) leurs chefs de clusters respectifs. Entre deux visites du collecteur, les deux clusters C1 et C2 se regroupent et forment un seul cluster C3 (avec ou sans d'autres nœuds n'appartenant préalablement ni à C1 ni à C2). Plusieurs cas peuvent se présenter selon le résultat de l'exécution de l'algorithme de clustering :

- Cas1 : CH(C1) devient chef de cluster du nouveau cluster C3. Donc CH(C3) = CH(C1)
- Cas2 : CH(C2) devient chef de cluster du nouveau cluster C3. Donc CH(C3) = CH(C2)

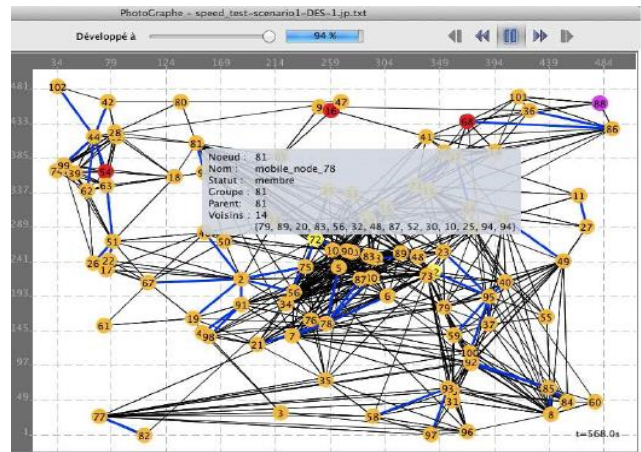


Figure 4 : L'outil « photographe »

- Cas3 : un ancien nœud membre (v) du cluster C1 ou C2 devient chef de C3. Donc CH(C3) = (v).
- Cas4 : Un nœud (u), qui n'appartenait ni à C1 ni à C2, devient chef de C3. Donc CH(C3) = (u).

Dans les cas 1 ou 2, le collecteur lors de sa prochaine visite dans la région trouvera CH(C3)=CH(C1) ou CH(C3)=CH(C2), et relèvera les informations relatives à la zone couverte par le cluster C3. Il faut pouvoir l'informer de l'évolution du contexte. Cela peut paraître assez simple. Mais dans les cas 3 et 4, le collecteur lors de sa prochaine visite au chef de cluster CH(C3)=(v) ou CH(C3)=(u) relèvera des informations relatives à la zone couverte par le cluster C3 pour lesquelles l'association avec le précédent contexte paraît moins évidente...

Dans les quatre cas, cités ci-dessus, le problème posé au niveau du collecteur est un problème de représentativité d'informations collectées par zone. Autrement dit, comment le collecteur peut savoir que les informations, qu'il vient de récupérer au niveau du nouveau CH(C3), représentent celles des zones C1 et C2 qu'il connaissait préalablement, sachant qu'il n'a aucune information géographique sur l'emplacement de ces zones, et que ses correspondants ont potentiellement changé ? La planification de sa trajectoire s'avère difficile. La figure 5 montre 2 autres scénarios.

On comprend donc la nécessité d'implémenter au niveau du collecteur un protocole de session qui lui permettra de récupérer le contexte de ses précédentes visites. Il pourra ainsi reprendre le dialogue avec l'ancien ou éventuellement le nouveau chef de cluster élu, regrouper les informations spécifiques à chaque zone et les classer dans le bon ordre, et pourra donc recréer le contexte global de la collecte des données. Autrement dit, même avec des connexions discrètes aux instants de visite du collecteur aux chefs de clusters, il doit pouvoir reconstruire un système d'information cohérent sur l'ensemble des connexions déjà établies. Par exemple, dans le cas d'une fusion de clusters où il y a changement de chef, le collecteur attribue à ce dernier un identifiant de session déduit des anciens

identifiants de session des anciens chefs des clusters fusionnés.

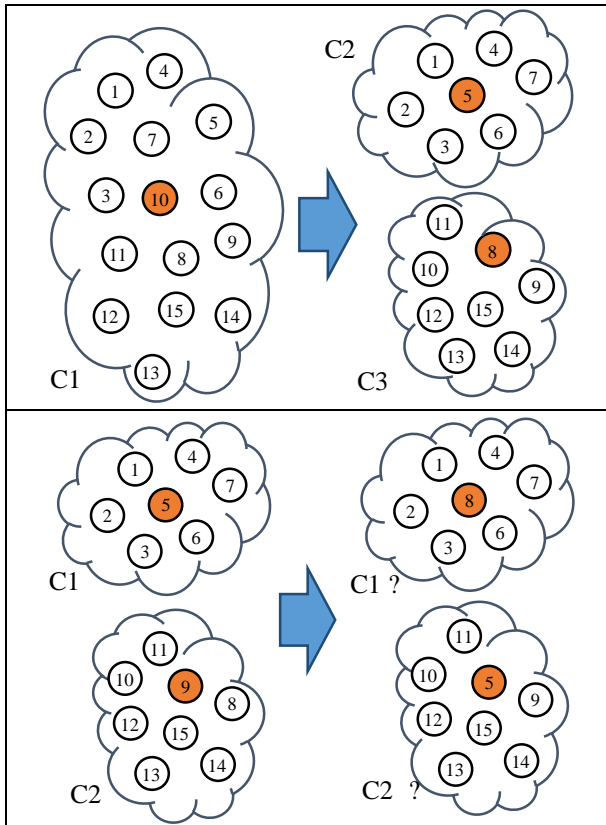


Figure 5 : Deux scénarios d'évolution de clusters

3.1 Définitions

La topologie du réseau change constamment suivant l'évolution de la dispersion des nappes de pétrole en mer. Ainsi, des groupes de nœuds capteurs peuvent fusionner et former un plus grand groupe. A l'inverse, un groupe de nœuds capteurs peut se diviser en deux ou plusieurs sous-groupes de nœuds capteurs. Ces groupes dynamiques sont de taille variable, et donc informationnellement significatifs ou non (y a-t-il intérêt à visiter un groupe de deux nœuds représentant une nappe de 20 cm de diamètre ?). Nous définissons donc la session et l'héritage, sur lesquels se basent les fusions ou les scissions qui représentent les différents cas de figure de la dynamique du réseau, et qui vont gouverner la stratégie de visite du (des) collecteur(s).

3.1.1 Session

Une session représente l'échange d'un sous-ensemble informationnel du système d'information global $S(k)$, entre un collecteur et un groupe de nœuds G à l'occasion de sa visite. Elle a donc une dimension spatiale et temporelle. k évolue à chaque nouvel échange du collecteur avec un groupe de nœud, et représente donc la discrétisation de la connectivité du réseau de capteurs global. Attention, on confondra maintenant la même notation, pour le groupe de nœuds G_i , et pour la session G_i qui représente le sous-

ensemble informationnel de $S(k)$ porté par le groupe G_i à l'instant k .

Dans un souci d'efficacité de la collecte, l'échange doit être informationnellement significatif. On peut donc l'associer à un groupe de nœuds (cluster) de taille suffisante ($|G| \geq n$). Un cluster trop petit ne fera donc pas l'objet d'une session. Il ne sera pas visité car informationnellement négligeable.

Une session représente donc l'échange d'information avec un cluster composé d'au moins n nœuds. La valeur du nombre n dépend du nombre total de nœuds, mais surtout de la finesse avec laquelle on veut observer le phénomène physique. Une grande valeur de n fournira une vision macroscopique de la réalité. Une petite valeur de n donnera une vision plus précise, mais compliquera considérablement le programme de visite du (des) collecteur(s).

L'association d'une session à un cluster permet de réduire la trajectoire du collecteur aux chefs des différents clusters. Nous considérons qu'à chaque visite du collecteur au chef de cluster représentant un groupe de nœuds G_i , ce dernier peut lui communiquer les paramètres suivants : l'ensemble des identifiants des membres de son groupe $\{G_i\}$, sa position $Pos(CH(G_i))$ et celle du nœud membre le plus éloigné par rapport au chef de cluster $Pos(Cmk)$, ainsi que les mesures brutes ou prétraitées (par exemple, la moyenne $Dmoy$ des mesures).

3.1.2 Héritage

Cette relation est essentielle, car elle va permettre le suivi du contexte entre deux visites du (des) collecteur(s), même si les clusters et/ou leurs chefs ont changé. En effet, un cluster peut avoir perdu ou gagné des nœuds, peut avoir élu un nouveau chef, peut s'être fondu dans un autre, ou avoir donné lieu à plusieurs autres clusters. Le collecteur doit pouvoir associer le nouveau contexte qu'il découvre à la situation qu'il avait observée lors de sa précédente visite.

$$G_i \in S(k+1) \text{ hérite de } G_j \in S(k) \\ \Leftrightarrow \exists g \subset G_i / g \subset G_j \text{ et } |g| \geq \xi \cdot n$$

ξ représente la variation du nombre de transfuges autour de la taille minimale pour former un groupe. Il est positif et peut être inférieur à 1 pour éviter de créer une nouvelle session lorsque des transfuges de différents groupes fusionnent pour en former un nouveau (double héritage) ou lorsqu'un groupe rétrécit.

3.1.3 Fusion

Une session est une fusion d'autres sessions si elle hérite d'au moins deux sessions. La fusion symbolise le cas où une partie significative de deux ou plusieurs groupes fusionnent pour créer un nouveau groupe.

3.1.4 Scission

Une session se scinde en deux ou plusieurs autres sessions si chacune hérite de la session d'origine à l'occasion de la

division du groupe auquel elle était associée en groupes de taille suffisante.

3.1.5 Création/suppression de session

Une session est considérée nouvellement créée si elle est associée à un groupe de taille suffisante et si elle n'hérite d'aucune session existante.

Une session disparaît si aucune session n'en hérite ou si le groupe auquel elle était associée ne contient plus au moins n nœuds.

3.2 Description du protocole

Comme il y a rupture de connectivité, il est important de noter qu'une session est organisée en deux phases. La première, dite de collecte, consiste à récupérer l'ensemble des informations mesurées sur le collecteur. L'utilisation de nœuds collecteurs mobiles a notamment été abordée pour l'amélioration du routage (Pazzi R.W.N. et Boukerche A., 2008) ou, comme dans ce papier, pour la récupération d'informations à l'instar de (Shah R.C., et al., 2003). La seconde phase consiste en un échange plus classique de couche session pour gérer des points de synchronisation entre le collecteur mobile et le système d'information. Le service offert par la couche session est défini dans le standard (ISOa, 1996) et est notamment implanté par les protocoles X.25 (ISOb, 1996), SCP ou SIP.

Comme mentionné dans (Nussbaumer H., 1991), les points de synchronisation servent à identifier des portions du flot de données dans le double but d'isoler les différentes parties logiques du dialogue et de permettre la reprise en cas d'erreur ou d'incident. Dans cet article, la notion d'incident sera associée à la rupture de connectivité du réseau, et le dialogue sera repris à chaque cycle du collecteur mobile. Les points de synchronisation majeurs servent à séparer complètement les parties de la communication qui le précède et qui le suit. Ils définissent une unité de dialogue. L'expéditeur ne peut entamer une nouvelle unité de dialogue sans avoir reçu de confirmation sur la précédente. Les points de synchronisation majeurs correspondent dans notre application aux photographies du réseau établies à chaque cycle du collecteur. Toujours selon (Nussbaumer H., 1991), les points de synchronisation mineurs permettent de structurer l'unité de dialogue. L'expéditeur peut continuer à envoyer des données après le point de synchronisation, sans attendre de confirmation. Dans notre protocole, les points de synchronisation mineurs séparent les échanges du collecteur avec chaque groupe de capteurs faisant l'objet d'une session.

A notre connaissance, peu de travaux se sont intéressés à la couche session pour des réseaux semblables à notre cas d'étude. (Demmer M. et Fall K., 2009) définissent une couche session pour réseaux DTN (Delay Tolerant Networks) avec comme objectif le support partiel de l'ordre des transactions (*bundle*), un modèle de service *publisher/subscriber/custodian* et la gestion des

obsolescences (des *bundles*). La gestion des contenus est traitée *via* un URL par groupe de *bundles* avec en complément un mécanisme d'estampillage pour apprécier l'évolution temporelle. En comparaison, un seul groupe peut avoir changé dans notre cas d'étude, et pas seulement sa valeur, mais aussi son périmètre comme défini au paragraphe 3.1.1. Par rapport à la gestion des rôles définie dans (Demmer M. et Fall K., 2009), le collecteur est à rapprocher du *custodian* (généralement un routeur qui agit comme DTN proxy) mais ici, la connexion réseau sera assurée par le déplacement du collecteur. Les points de synchronisation majeurs dans (Demmer M. et Fall K., 2009) sont enfin associés à un numéro de séquence (pour la gestion de l'ordre des paquets sur un réseau, sachant que dans notre cas d'étude, on revient à une simple liaison point-à-point) et à un estampillage qui suppose une synchronisation d'horloge que nous ne considérons pas.

Nous nous intéresserons dans cet article à une couche session (point-à-point) avec connexion (ISOa, 1996) et donc aux mécanismes de gestion de connexion associés, mais n'abordons pas la gestion des conflits. La figure 8 donne un aperçu des différents échanges et interactions entre les constituants de l'architecture, de la mesure physique sur un capteur, à la contextualisation (spécification de la dynamique) des données au sein du système d'information. Les échanges relatifs au partitionnement ne sont pas reportés ici.

3.2.1 Echanges pour la collecte

La première fonction du collecteur mobile consiste à récupérer les mesures réalisées par l'ensemble des nœuds. Comme évoqué précédemment, plutôt que d'aller voir tous les capteurs, le collecteur se contente de visiter un chef de cluster représentatif d'une « nappe ». La production de cette mesure est décrite par le diagramme d'état « enrichi » d'un capteur donné par la Figure 6.

Les états (en jaune) ordinaire, chef, membre1 et membre 2 correspondent au processus de clustering (voir figure 3). Pour les trois derniers, le capteur mesure périodiquement (période τ) et émet une mesure *data* (événement d) vers son chef de cluster. Au niveau d'un chef de cluster, pour limiter la consommation des ressources (mémoire et processeur) et limiter *in fine* la consommation d'énergie, le traitement (fonction *updateData()*) des données reçues (événement d) est réduit au stockage d'une liste des capteurs membres du cluster $\{G_i\}$, de la surface du cluster ($Pos(CH)$ et $Pos(Cmk)$) et au calcul d'une donnée agrégée (par exemple D_{moy} si le chef effectue un prétraitement sur les données mesurées en calculant la moyenne). Cette agrégation introduit une perte de contexte puisque le cluster (la nappe) est dynamique.

Au début d'un cycle (*débutCycle()*), le collecteur, après *déplacement()*, envoie un message *connect()* permettant d'initier une session de récupération des données auprès d'un chef de cluster. Ensuite, il transmet un message *S_get-data()* auquel le chef répond en renvoyant la mesure agrégée pour son cluster (D_{moy}).

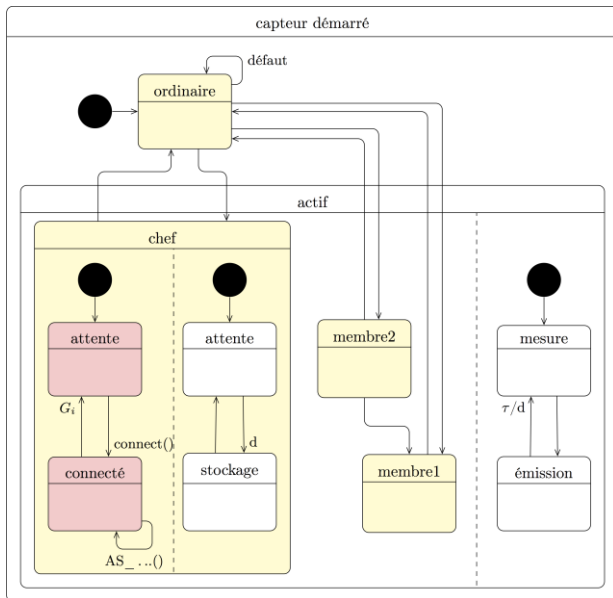


Figure 6 : Nouveau diagramme d'état d'un nœud capteur

Le collecteur demande enfin par un message $S_{get-context}()$ le contexte du cluster. Le chef transmet ainsi les variables $\{G_i\}$, $Pos(CH)$ et $Pos(Cmk)$, ce qui met fin à la session entre le collecteur et le cluster. Le dernier message reçu permet alors au collecteur d'exécuter via la fonction $updateSessions()$ les tests d'héritage définis ci-avant et recouvrer ainsi l'évolution du cluster et identifier la perte de contexte correspondante. Cet enchaînement, déplacement et visite, est répété par le collecteur pour l'ensemble des chefs de cluster (figure 7). Le collecteur aura ainsi d'une vue de la dynamique du réseau et mettra à jour sa trajectoire pour atteindre les différents groupes de nœuds suffisamment représentatifs. La trajectoire de nœuds mobiles pourra ensuite être contrôlée comme dans (Wang G.G. et al., 2007).

3.2.2 Echanges de session

La couche session a pour objectif de fournir aux entités collecteur et système d'information (SI) en charge de la présentation des données, les moyens pour la synchronisation de leur dialogue et la contextualisation de l'information. Le collecteur définit des points (de synchronisation) de reprise de la mise à jour des données (message $S_{sync}(t)$). L'intervalle de reprise n'est pas forcément périodique, et plus le cycle sera court, plus on se rapprochera d'une connectivité « continue ». La connexion sera fermée par un message $S_{finish}()$ du collecteur, après confirmation $disconnect$ du SI.

Pour chaque groupe pour lequel le collecteur a récupéré des informations, une activité qui correspond à une synchronisation locale est initiée. Trois types d'activité sont définis. La première est représentative de l'apparition d'une nouvelle nappe, et donc d'un nouveau groupe. À la réception d'un message $S_{session-start}()$, le SI crée ($create()$) dans sa base locale un nouvel objet session, et retourne son identifiant (sei) au collecteur.

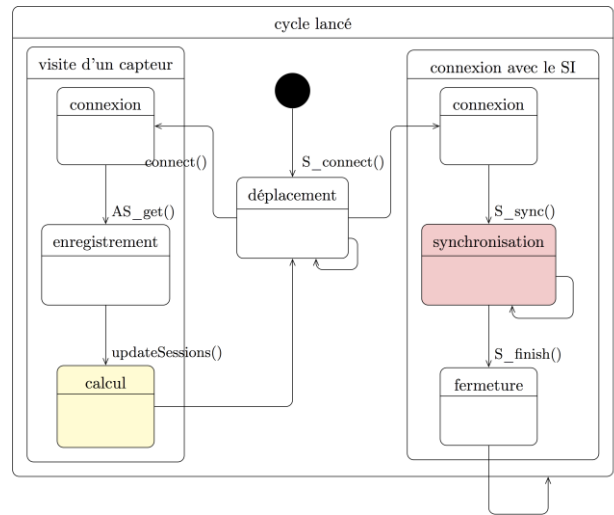


Figure 7 : Diagramme d'état du collecteur mobile

Cet identifiant permet à celui-ci d'y associer les données et leur contexte (message $S_{data}()$) avant de suspendre la session par un message $S_{session-interrupt}()$.

Une session suspendue sera reprise suite à la réception d'un message $S_{session-resume}()$ comme le montre la Figure 9.

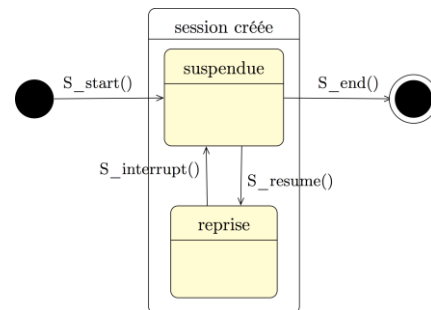


Figure 9 : Diagramme d'état d'un objet session

Cela permettra ainsi au collecteur de mettre à jour les attributs de la session (message $S_{data}()$), qui sera de nouveau suspendue jusqu'au cycle suivant. C'est à cette activité, que les relations d'héritage obtenues par comparaison entre deux points de synchronisation majeurs (ici entre deux cycles), sont marquées.

Enfin, compte tenu de la mobilité des nappes, et des éventuelles scissions ou fusions de nappe, le collecteur pourra demander (message $S_{session-end}()$) au SI, l'écartement d'une session. Dans ce cas, le SI retirera l'objet associé ($discard()$) avant de confirmer sa suppression (message ack).

Les échanges de session sont asynchrones aux cycles de production de la mesure et d'auto-organisation (clustering) du réseau. Ils sont déclenchés après le parcours par le collecteur de la zone à surveiller et permettent d'obtenir, de manière transparente au SI, une connectivité discrète « historisée ». En effet, ce protocole permet d'établir pour chaque image des nappes (obtenue à chaque cycle), des relations avec les images précédentes. On obtiendra

ainsi une représentativité spatiale (position et taille des groupes) et temporelle (relations entre les groupes). Cette dernière pourra par ailleurs être utilisée pour un pronostic sur la trajectoire des nappes (et donc du collecteur).

4 SIMULATIONS

Pour montrer le bon comportement du protocole proposé, nous avons effectué des séries de simulations sous Riverbed Opnet Modeler. L'objectif est d'évaluer le protocole selon trois métriques : le nombre de sessions, le nombre de nœuds hors session et le nombre de nouvelles sessions entre deux visites du collecteur.

- le nombre de sessions définit le nombre de clusters significatifs à visiter. Tout cluster d'au moins n membres est une session.

- le nombre de nœuds hors sessions nous permet d'estimer l'aspect statique de notre réseau à un instant donné, et donc la capacité de couverture de toute la zone de captage. Plus le nombre de nœuds hors session est faible, plus la zone de captage est bien couverte.

- le nombre de nouvelles sessions entre deux visites du collecteur, nous permet d'estimer l'aspect dynamique du réseau. Plus il est faible, plus le suivi de l'évolution des nappes est précis, donc meilleure est la prédiction des futures positions des chefs de clusters, ce qui facilitera la planification de la trajectoire future du collecteur.

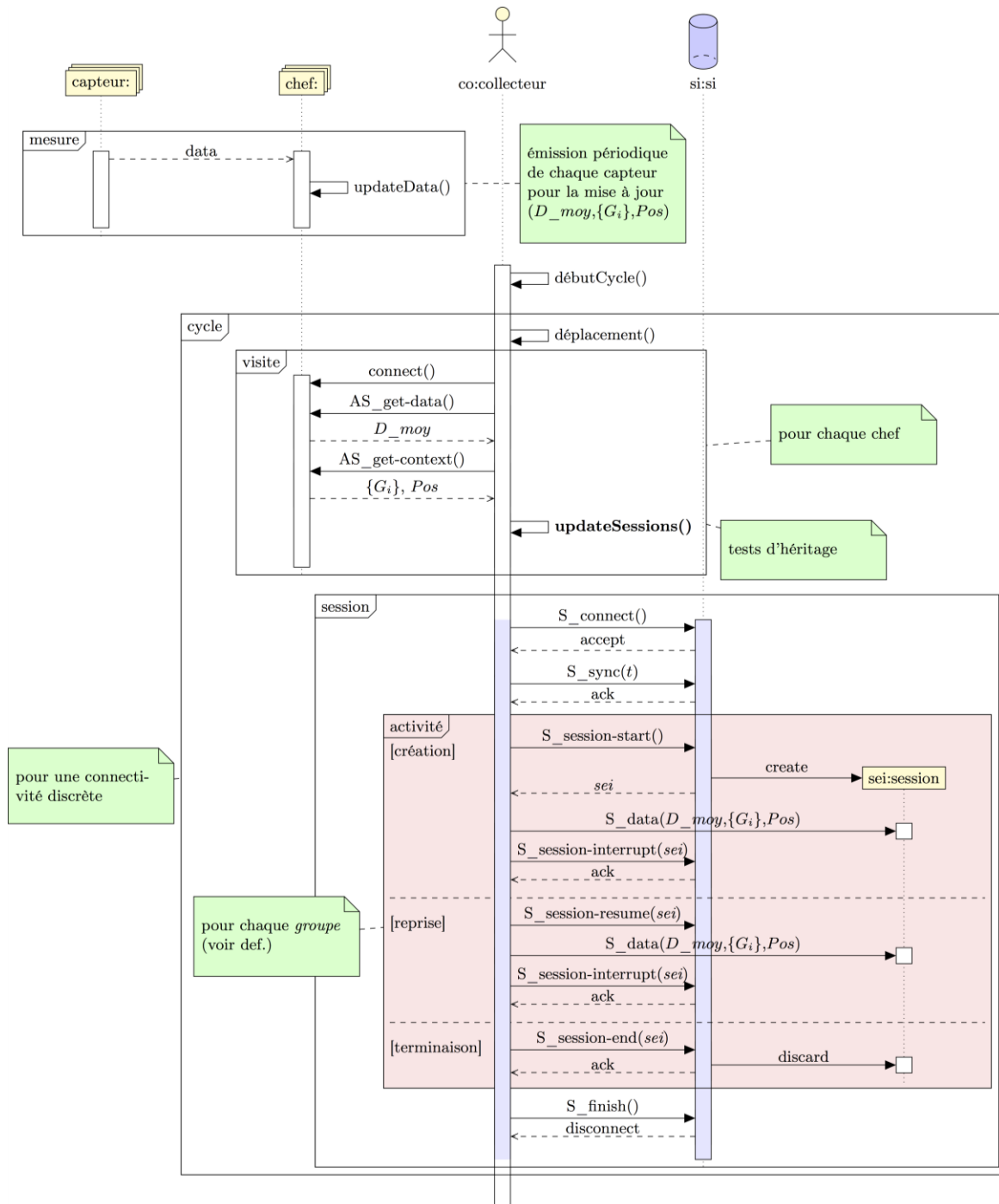


Figure 8 : Diagramme de séquence des échanges sur le réseau de capteurs.

On fera varier au cours des simulations le nombre de nœuds représentatifs d'une session, la densité des nœuds, la vitesse des nœuds, et le paramètre ξ qui représente la variation du nombre de transfuges autour de la taille minimale pour former un groupe.

4.1 Paramètres de simulation

La surface observée est limitée afin d'avoir des temps de calcul des simulations qui restent raisonnables, et éviter l'usage de machines surpuissantes dotées de mémoire vive importante. Le nombre de nœuds nécessaires à un bon monitoring dépend de leur portée. Un ordre de grandeur de 30 m conduit à prendre 500 nœuds pour une surface de 625 x 625 m. Les paramètres considérés sont :

| Paramètres | Valeurs |
|--|---|
| Nombre de nœuds | 500 |
| Surface observée | 625 m x 625 m |
| Pondération pour clustering (α) | 0,5 |
| Durée de la simulation | 300 s |
| Nombre de seeds/simulation | 30 |
| Période des messages <i>Hello</i> | 1 s |
| Période de visite du collecteur | 5 s |
| Taille mini d'une session (n) | 10, 25 et 50 nœuds |
| Vitesse des nœuds | 1 et 2 m/s |
| Modèle de mobilité | Random Waypoint |
| Nombre de simulations | $4 \times 3 \times 2 \times 3 \times 30 = 2160$ |

Tableau 1: Valeurs des paramètres de simulation

4.2 Nombre de sessions

La figure 10 présente, à $V=1\text{m/s}$, l'évolution du nombre de sessions en fonction du temps pour différentes valeurs du nombre n ($n \in \{10, 25, 50\}$). On remarque que plus n est grand, plus le nombre de sessions est petit, ce qui est parfaitement logique. Ces courbes ont été obtenues avec $\xi = 100\%$. Elles ne changent pas si on prend successivement $\xi = 50, 150$ ou 200% . Ce paramètre n'a donc aucun effet sur le nombre de sessions. A $V = 2\text{m/s}$, l'allure des courbes est identique.

4.3 Nombre de nœuds hors session

La figure 11 présente, à une vitesse de déplacement $V=1\text{m/s}$, l'évolution du nombre de nœuds hors sessions en fonction du temps pour différentes valeurs du nombre n ($n \in \{10, 25, 50\}$). On remarque que plus n est grand, plus le nombre de nœuds hors sessions est grand, ce qui est normal, puisqu'à vouloir de « grosses » sessions, on néglige de nombreux « petits » clusters. L'observation perd donc de sa finesse quand n grandit.

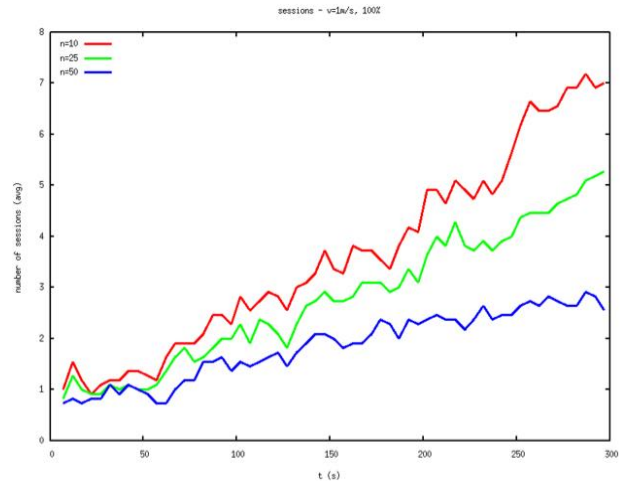


Figure 10 : Evolution dans le temps du nombre moyen de sessions pour $n \in \{10, 25, 50\}$

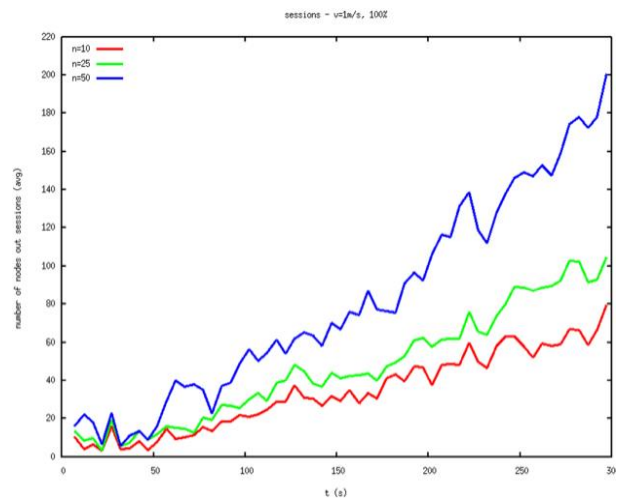


Figure 11 : Evolution dans le temps du nombre moyen de nœuds hors sessions pour $n \in \{10, 25, 50\}$

Ces courbes ont été obtenues avec $\xi = 100\%$. Elles ne changent pas si on prend successivement $\xi = 50, 150$ ou 200% . Ce paramètre n'a donc aucun effet sur le nombre de nœuds hors sessions. A $V = 2\text{m/s}$, l'allure des courbes est identique.

4.4 Nombre de nouvelles sessions

La figure 12 présente, à une vitesse de déplacement de 1m/s , l'évolution du nombre de nouvelles sessions en fonction du temps pour différentes valeurs du nombre n ($n \in \{10, 25, 50\}$).

A petite vitesse, et pour toutes valeurs de ξ (100% sur la figure), les courbes ne permettent pas de dégager une tendance permettant de statuer sur l'influence des paramètres. Par contre, à grande vitesse, les courbes sont plus expressives et influencées par le paramètre ξ .

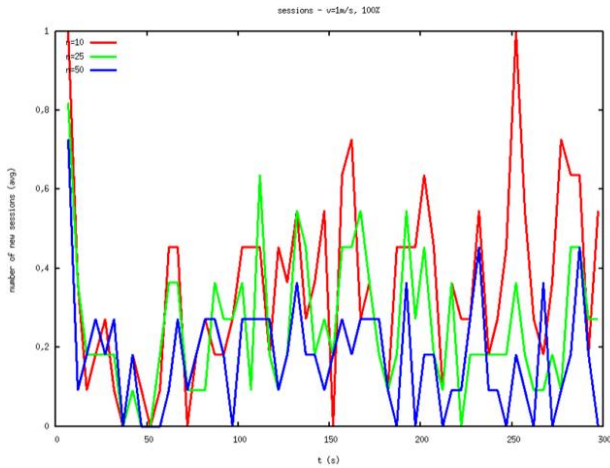


Figure 12 : Evolution dans le temps du nombre moyen de nouvelles sessions pour $V=1m/s$ et $n \in \{10, 25, 50\}$

Les figures 13 à 16 présentent à une vitesse de déplacement de 2m/s, respectivement pour $\xi = 50, 100, 150$, et 200%, l'évolution du nombre de nouvelles sessions en fonction du temps pour différentes valeurs du nombre n ($n \in \{10, 25, 50\}$).

Pour des valeurs du facteur ξ égales à 50% et 100%, le nombre de nouvelles sessions créées augmente dans le temps pour des valeurs de n petite et diminue pour les grandes valeurs de n . Ce qui représente le résultat recherché c'est à dire une faible perte de contexte, permettant au collecteur de retrouver des sessions « connues » d'une visite à l'autre.

Pour de grandes valeurs du facteur ξ (150% et 200%), nous remarquons que le nombre des nouvelles sessions diminue dans le temps, même pour des valeurs de n , mais plus vite que précédemment, car ξ possède un caractère amplificateur de la difficulté d'hériter de nappes de plus en plus petites...

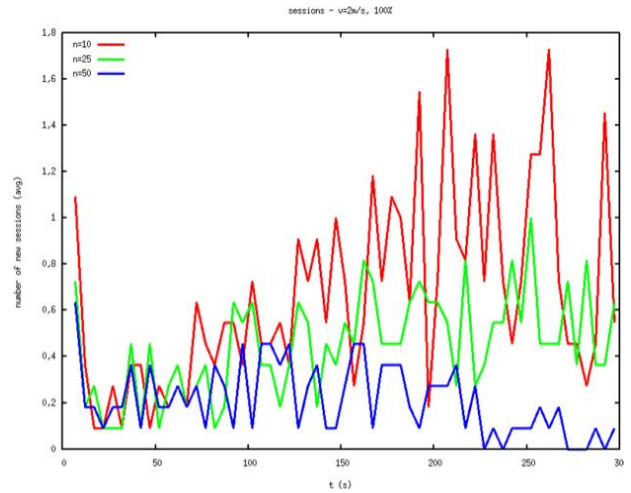


Figure 14 : Evolution dans le temps du nombre moyen de nouvelles sessions pour $V=2m/s$, $\xi = 100\%$, et $n \in \{10, 25, 50\}$

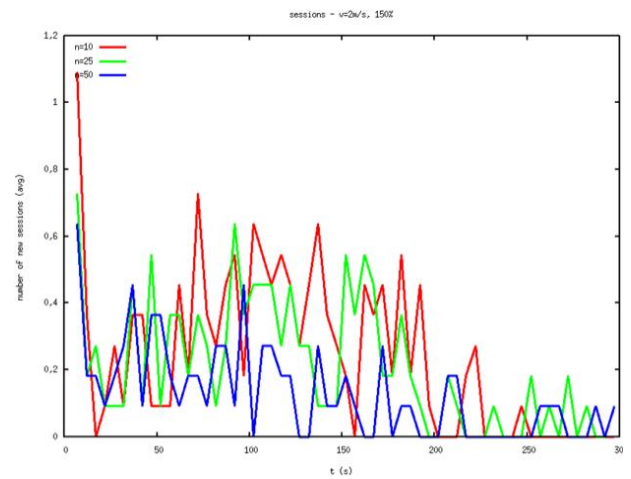


Figure 15 : Evolution dans le temps du nombre moyen de nouvelles sessions pour $V=2m/s$, $\xi = 150\%$, et $n \in \{10, 25, 50\}$

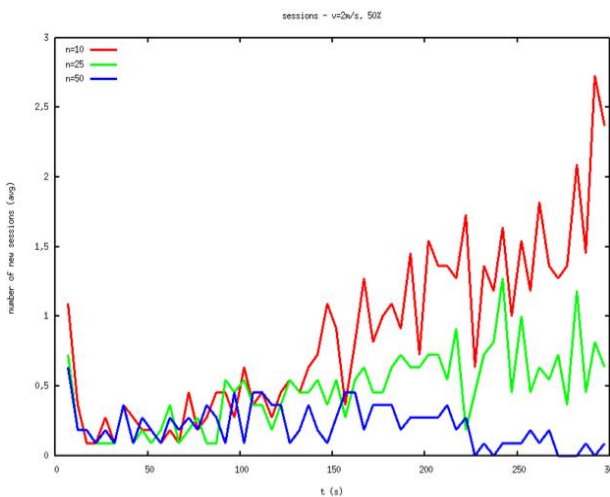


Figure 13 : Evolution dans le temps du nombre moyen de nouvelles sessions pour $V=2m/s$, $\xi = 50\%$, et $n \in \{10, 25, 50\}$

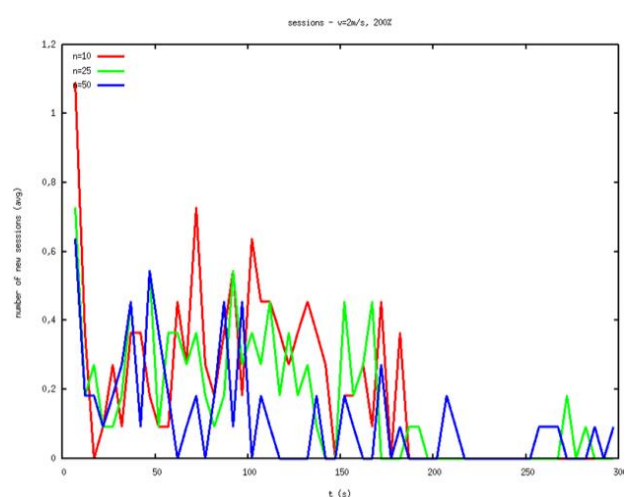


Figure 16 : Evolution dans le temps du nombre moyen de nouvelles sessions pour $V=2m/s$, $\xi = 200\%$, et $n \in \{10, 25, 50\}$

5. DISCUSSION

Nous avons constaté que plus n est grand, plus le nombre de nœuds hors sessions augmente, ce que nous cherchons à éviter, puisque l'écart entre monde réel et virtuel augmente par non prise en compte d'une partie importante des capacités de mesure du réseau de capteurs. Nous avons constaté aussi que quand ξ est petit, le nombre de nouvelles sessions a tendance à augmenter, ce que nous cherchons pourtant à éviter, puisqu'elles représentent une perte de contexte, et donc une difficulté de planification de la trajectoire du collecteur à chacune de ses « tournées ». Comme un groupe n n'hérite que si sa taille satisfait $|g| \geq \xi \cdot n$, il faut trouver un compromis entre le suivi statique du phénomène physique porté par n , et le suivi de la dynamique de ce même phénomène porté par ξ . Pour gérer cette contradiction l'exploitant du système d'information devra non seulement alimenter le simulateur avec les paramètres correspondant à la situation précise à surveiller (étendue de la marée noire, conditions météorologiques, ...), mais aussi raffiner ses exigences en termes de finesse d'observation (taille de plus petite nappe que l'on souhaite observer), pourcentage du nombre total de nœuds du réseau dont on veut prendre en compte les mesures, ..., et enfin définir les périodes de visite, la tolérance à la quantité d'information non contextualisée, ...

Tout ce paramétrage préalable permettra de trouver par simulation des valeurs optimales de n et ξ pour une catastrophe donnée.

CONCLUSION

A notre connaissance, cette proposition est tout à fait originale et constitue une réelle nouveauté dans le domaine des réseaux de capteurs. Elle se distingue des travaux menés sur les DTN (Delay Tolerant Network) (Fall K., 2003), (Demmer M. et Fall K., 2009) ou sur les datamules (Jea D. et al., 2005). En effet, au-delà de la simple collecte des données, elle prend en compte le contexte informationnel que l'on cherche à préserver au fur et à mesure des visites successives opérées par le collecteur.

Les travaux futurs devront porter sur :

- l'étude de la consommation énergétique de notre protocole de session qui doit rester raisonnable, car celui-ci est en partie mis en œuvre sur des nœuds à capacité limitée ;
- l'usage de modèles de mobilité réalistes intégrant séparément mobilité du groupe (nappes) et mobilité individuelle (nœud) ;
- le déploiement simultané de plusieurs collecteurs mobiles.

REFERENCES

Abbasi A.A. and Younis, M., 2007. A survey on clustering algorithms for wireless sensor networks. *Computer Communications*, vol 30, pp.2826-41.

- Arboleda L.M.C. and Nasser N., 2006. Comparison of Clustering Algorithms and Protocols for Wireless Sensor Networks, *Canadian Conference on Electrical and Computer Engineering*, Ottawa.
- Demmer M. and Fall K., 2009. The design and implementation of a session layer for delay-tolerant networks. *Computer Communications*, vol 32, pp. 1724-1730.
- Fall K., 2003. A Delay-tolerant Network Architecture for Challenged Internets. *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Karlsruhe, Germany, pp. 27-34.
- Harchi S., Georges J.-P., and Divoux T., 2012. WSN dynamic clustering for oil slicks monitoring. *3rd International Conference on Wireless Communications in Unusual and Confined Areas, ICWCUCA'12*, Clermont-Ferrand, France.
- ISOa, 1996. Information technology -- Open Systems Interconnection -- Session service definition, ISO/IEC 8326, 1996.
- ISOb, 1996. Information technology -- Open Systems Interconnection -- Connection-oriented Session protocol: Protocol specification, ISO/IEC 8327-1, 1996.
- Jea D., Somasundara A., Srivastava M., 2005. Multiple controlled mobile elements (data mules) for data collection in sensor networks, *Proceedings of the First IEEE international conference on Distributed Computing in Sensor Systems DCOSS'05*, Marina del Rey, California, pp. 244-257.
- Kumarawadu P., Dechene D.J., Luccini M. and Sauer A., 2008. Algorithms for Node Clustering in Wireless Sensor Networks: A Survey, *4th International Conference on Information and Automation for Sustainability*, Colombo, Sri Lanka.
- Levy E. M., 1979. The IGOSS Pilot Projet on Marine Pollution (Petroleum) Monitoring: Its Evolution and a Personal Viewpoint. *Marine Pollution Bulletin*, Vol 10, Issue 1, pp. 5-11.
- Mitton, N., 2006. Auto-organisation des réseaux de capteurs sans fil multi-sauts à grande échelle. *Thèse de l'INSA de Lyon*.
- Nussbaumer, H., 1991. Téléinformatique III. *Presses Polytechniques et Universitaires Romandes*, Collection Informatique. ISBN 2-88074-204-8.
- Pazzi R.W.N. and Boukerche A., 2008. Mobile data collector strategy for delay-sensitive applications over wireless sensor networks. *Computer Communications*, Vol. 31, pp. 1028-1039.
- Shah R.C., Roy S., Jain. S. and Brunette, W., 2003. Data MULEs: modeling a three-tier architecture for sparse sensor networks. *Ad-hoc Networks*, Vol. 1, Issue 2-3, pp. 215-233.
- Wang G.G., Cao G., Berman P. and La T.F., 2007. Bidding protocols for deploying mobile sensors. *IEEE Transactions on Mobile Computing*, Vol. 6, issue 5, pp. 563-576.