Algorithmique répartie adaptative

Bertrand Ducourthial

UMR CNRS 6599 HEUDIASYC

Université de Technologie de Compiègne

Bertrand.Ducourthial@hds.utc.fr

Plan

- Définition
  - du système réparti au système réparti adaptatif
  - choix des adaptations
  - implications

Plan

- Définition
- Objectifs
  - qui est mobile ?
  - limitations... (hostilité, complexité)
  - recherche d’une application type

Plan

- Définition
- Objectifs
- Modèles
  - mobiles, communications, réseau
  - implications, uniformité ? / “généricité” ?
  - où situer les développements ? démarche

Plan

- Définition
- Objectifs
- Modèles
- Techniques algorithmiques
  - utilisation du routage
  - épine dorsale
  - point d'ancrage
  - agents
  - auto-stabilisation
  - détecteurs de défaillances
  - mobile $=$ message persistent

Plan

- Définition
- Objectifs
- Modèles
- Techniques algorithmiques
- Conclusion

Système réparti adaptatif

- Système réparti $\mathcal{S}$
  - entités qui calculent et communiquent
  - réparties dans l'espace
  - entité = plus petite unité produisant un calcul *unité de calcul*

Système réparti adaptatif

- Système réparti $\mathcal{S}$
  - entités qui calculent et communiquent
  - réparties dans l'espace
  - entité = plus petite unité produisant un calcul
    unité de calcul
  - système $\equiv$ ce dont on est sûr, hypothèses
    - réseau modélisé par un graphe
    - connaissance globale
      e.g., identifiants, voisinage, horloge, . . .
    - *etc.*
  - on est seul au monde

Système réparti adaptatif

- Système réparti $\mathcal{S}$
- Dans la réalité
  - le système $\mathcal{S}$ barbotte dans un système plus vaste $\overline{\mathcal{S}}$
  - modélisation d'une sous-partie de $\overline{\mathcal{S}}$
  - influence du système global sur le sous-système

Système réparti adaptatif

- Système réparti $\mathcal{S}$
- Dans la réalité
  - le système $\mathcal{S}$ barbotte dans un système plus vaste $\overline{\mathcal{S}}$
  - modélisation d'une sous-partie de $\overline{\mathcal{S}}$
  - influence du système global sur le sous-système

  - le système $\mathcal{S}$ peut évoluer
  $\rightsquigarrow$ les hypothèses évoluent
  $\rightsquigarrow$ erreur dans la modélisation ?
  - que reste-t-il de constant ?

Système réparti adaptatif

- Système réparti $\mathcal{S}$
- Dans la réalité
- Système adaptatif
  - adapter les algorithmes de $\mathcal{S}$ aux perturbations de $\overline{\mathcal{S}}$
  - adapter les algorithmes de $\mathcal{S}$ aux modifications de $\mathcal{S}$ par lui-même
  - modifications de $\mathcal{S}$
    - par l'intérieur $\mathcal{S}$
    - par l'extérieur $\overline{\mathcal{S}} \setminus \mathcal{S}$

$\rightsquigarrow$ système qui «réagit», qui «s'adapte»

Système réparti adaptatif

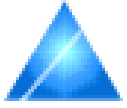
- Système réparti $\mathcal{S}$
- Dans la réalité
- Système adaptatif
- À quoi s'adapter ?
  - ce type de simplication ne date pas d'aujourd'hui
  - choix, délimitation du domaine d'étude

À quoi s'adapter ?

- Applications sensibles au trafic
  - changements dans le volume ou la nature du trafic. . .
  - charge réseau en nombre de connexions, d'utilisateurs. . .
  - exemples :
    - montée en charge du système en fonction de l'heure, de la période
    - différentes matrices de routage des flux dans les réseaux
    - changement du site de téléchargement, de borne de diffusion, *etc.*

À quoi s'adapter ?

- Applications sensibles au trafic
- Applications sensibles aux pannes
  - fiabilité des communications
    - perte
    - duplication
    - modification
    - désordonnancement (?)



À quoi s'adapter ?

- Applications sensibles au trafic
- Applications sensibles aux pannes
  - fiabilité des communications
  - panne d'un site au démarrage
  - panne *crash* d'un site
  - pannes temporaires
    - corruption mémoire, message
    - entité déconnectée puis reconnectée
  - panne byzantine

À quoi s'adapter ?

- Applications sensibles au trafic
- Applications sensibles aux pannes
- Applications sensibles aux «agressions»
  - données / ressources à protéger / partager
  - identification des participants
  - coopération / arbitre

À quoi s'adapter ?

- Applications sensibles au trafic
- Applications sensibles aux pannes
- Applications sensibles aux «agressions»
- Applications sensibles à la dynamique des entités
  - arrivées / départs d'entités
  - identification, droit à participer ?
  - exemple : tables de routage vs. circulation jeton

À quoi s'adapter ?

- Applications sensibles au trafic
- Applications sensibles aux pannes
- Applications sensibles aux «agressions»
- Applications sensibles à la dynamique des entités
- Applications sensibles aux déplacements des entités
  - déplacement de terminaux, d'utilisateurs, de code
  - dépend du type de communication, de déplacement
  - notion de référentiel, localisation

Choix des adaptations

- Dans le cadre de l'AS Dynamo
  - graphes dynamiques
    i.e., adaptation aux changements de topologie
  - création / disparition / déplacement de noeuds

⇝ laisser de côté les perturbations extérieures ?
- adaptation à la charge, au trafic, *etc.*
- adaptation aux agressions
- . . .

Choix des adaptations

- Dans le cadre de l'AS Dynamo
- Qu'est-ce que l'«extérieur» ?
  - arrivée de nouveaux participants ?
  - trafic généré par le système lui-même ?
    - graphe logique mouvant dans un réseau physique fixe
    - *e.g.*, forêt des meilleurs distributions multisources, en fonction d'un critère, type panne, taux de pertes, ou charge ?
  - ignorer $\overline{\mathcal{S}} \setminus \mathcal{S}$?

Choix des adaptations

- Dans le cadre de l'AS Dynamo
- Qu'est-ce que l'«extérieur» ?
- Parti pris
  - adaptation à la dynamique des entités
    création / disparition
    $\rightsquigarrow$ un futur participant p n'est pas un extérieur :
    $p \notin \overline{\mathcal{S}} \setminus \mathcal{S}$
    cf. panne crash vs. apparition *from scratch*
  - adaptation aux déplacements
    $\rightsquigarrow$ le graphe n'est plus dans le modèle…

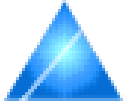
Choix des adaptations

- Dans le cadre de l'AS Dynamo
- Qu'est-ce que l'«extérieur» ?
- Parti pris
  - adaptation à la dynamique des entités
    création / disparition
    $\rightsquigarrow$ un futur participant p n'est pas un extérieur :
    $p \notin \overline{\mathcal{S}} \setminus \mathcal{S}$
    cf. panne crash vs. apparition *from scratch*
  - adaptation aux déplacements
    $\rightsquigarrow$ le graphe n'est plus dans le modèle...

  \+ pannes, sécurité, autres perturbations internes...

Dynamique de la topologie

- Mobilité et dynamique des entités
  - mobilité des participants
  - arrivées et départs
  - déplacement $\neq$ disparition puis (re-)création
    e.g., diffusion optimale



Dynamique de la topologie

- Mobilité et dynamique des entités
- Implications
  - prendre «conscience» de l'arrivée, la signaler
  - prendre «conscience» du départ, le signaler (!)
    $\rightsquigarrow$ panne ?
  - prendre «conscience» du déplacement
    - volontaire ?
    - détecter / signaler
    - référentiel, localisation
    - relatif / absolu *cf.* horloges

Dynamique de la topologie

- Mobilité et dynamique des entités
- Implications
- Mobilité / dynamique : nouveaux challenges
  - mobilité : caractéristique à part
    $\neq$ sans fil... (*cf.* mobilité des processus)

    challenge : trouver le destinataire,
    maintenir la communication
    malgré le déplacement
  - dynamique : caractéristique à part

Dynamique de la topologie

- Mobilité et dynamique des entités
- Implications
- Mobilité / dynamique : nouveaux challenges
  - mobilité : caractéristique à part
  - dynamique : caractéristique à part
    $\neq$ mobilité, $\neq$ sans fil...

    challenge : action de début en cours de route, accepter le nouveau participant, remise en cause résultat ou communication du résultat

Dynamique de la topologie

- Mobilité et dynamique des entités
- Implications
- Mobilité / dynamique : nouveaux challenges
  - mobilité : caractéristique à part
  - dynamique : caractéristique à part

  $\rightsquigarrow$ nouveaux algorithmes et techniques de développement

Plan

- Définition
- Objectifs
  - qui est mobile ?
  - limitations... (hostilité, complexité)
  - recherche d’une application type
- Modèles
- Techniques algorithmiques
- Conclusion

Qui est mobile ?

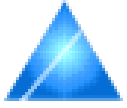
- *Unité de mobilité*
  - physique : terminaux
  - «humain» : utilisateurs
  - logique : code ?

  - *nomadic computing* : les utilisateurs bougent
  - *pervasive computing* : terminaux + utilisateurs
  - agents mobiles : code
    $\rightsquigarrow$ moyen et non finalité

Qui est mobile ?

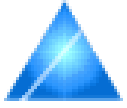
- *Unité de mobilité*
- À propos du code mobile
  - Agent mobile
    - pas d'obligation à résoudre un problème d'algorithmique répartie *via* des agents.
    - ce n'est pas une demande
    - c'est une technique de *design* parmi d'autres, mais pas la seule
    - appropriée dans certains cas
      e.g., travail en mode connecté/déconnecté

  $\rightsquigarrow$ un moyen et non une fin



Qui est mobile ?

- *Unité de mobilité*
- À propos du code mobile
- Qui est intrinsèquement mobile ? et dynamique ?
  - terminaux
  - utilisateurs
  - agents : un moyen parmi d'autres



Qui est mobile ?

- *Unité de mobilité*
- À propos du code mobile
- Qui est intrinsèquement mobile ? et dynamique ?

⇝ Nécessité de se définir des objectifs

- but de l'algorithmique répartie adaptative ?
- *killers apps*
- elles sont contraintes par l'hostilité de l'environnement et la complexité admissible par l'utilisateur

Hostilité de l'environnement

- *threads* au sein d'un processus
  - un seul programmeur
  - organisation coopérative
  - le programmeur peut laisser la main à l'un de ses *threads*, suivant l'ordonnanceur utilisé

Hostilité de l'environnement

- *threads* au sein d'un processus
- Processus dans un système multi-utilisateur
  - partage des ressources entre utilisateurs
  - travail coopératif ?
  - arbitre = OS

Hostilité de l'environnement

- *threads* au sein d'un processus
- Processus dans un système multi-utilisateur
- Routeurs dans un réseau
  - service offert à plusieurs utilisateurs
  - travail coopératif ?
  - un seul gestionnaire du réseau
  - pas d'accès des utilisateurs, organisation hiérarchique

Hostilité de l'environnement

- *threads* au sein d'un processus
- Processus dans un système multi-utilisateur
- Routeurs dans un réseau
- Accès à un médium de communication
  - partage
  - risque = capacité de nuisance

Hostilité de l'environnement

- *threads* au sein d'un processus
- Processus dans un système multi-utilisateur
- Routeurs dans un réseau
- Accès à un médium de communication
- Système à participants dynamiques
  - quel rôle a le participant ? confiance accordée ? *e.g.*, routage vs. exclusion mutuelle par jeton
  - qui arrive ? identification...

Hostilité de l'environnement

- *threads* au sein d'un processus
- Processus dans un système multi-utilisateur
- Routeurs dans un réseau
- Accès à un médium de communication
- Système à participants dynamiques
  - quel rôle a le participant ? confiance accordée ?
    e.g., routage vs. exclusion mutuelle par jeton
  - qui arrive ? identification...
  - certification ?
    nécessite une infrastructure fixe ($\neq$ adhoc)
  - filtrage ?
    toujours incomplet

Hostilité de l'environnement

- *threads* au sein d'un processus
- Processus dans un système multi-utilisateur
- Routeurs dans un réseau
- Accès à un médium de communication
- Système à participants dynamiques
- Point de vue
  - peu de chance que tout finisse connecté
  - cloisonnement
  - coopération *et* arbitres / hiérarchie

Règles d'usages

- Limitations humaines
  - applications (trop ?) futuristes
  - si utilisateur = homme, alors limitations

Règles d'usages

- Limitations humaines
- Exemple : partage de données
  - données partagés (*e.g.*,. un document)...
  - modifications concurrentes
  - l'informatique aide-t-elle ?

Règles d'usages

- Limitations humaines
- Exemple : partage de données
  - données partagés (*e.g.*,. un document)...
  - modifications concurrentes
  - l'informatique aide-t-elle ?
  - outils de contrôle de versions
  - mais en cas de conflits, réparations à la main.

Règles d'usages

- Limitations humaines
- Exemple : partage de données
- Exemple : Parker et perte connexité
  - un vecteur V_a par donnée partagée a
  - modification par le site $S_i \quad \rightsquigarrow \quad V_a[i]{+}{+}$
  - perte de connexité $\quad \rightsquigarrow \quad a', V'_a \quad$ et $\quad a'', V''_a$
  - après reconnexion, comparaison des vecteurs

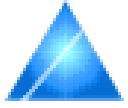
$$V'_a \preceq V''_a \quad \Rightarrow \quad a \leftarrow a'$$
$$V'_a \parallel V''_a \quad \Rightarrow \quad a =?$$

Règles d'usages

- Limitations humaines
- Exemple : partage de données
- Exemple : Parker et perte connexité
  - un vecteur V_a par donnée partagée a
  - modification par le site $S_i \quad \rightsquigarrow \quad V_a[i]{+}{+}$
  - perte de connexité $\quad \rightsquigarrow \quad a', V'_a \quad$ et $\quad a'', V''_a$
  - après reconnexion, comparaison des vecteurs

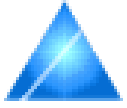
    $V'_a \preceq V''_a \quad \Rightarrow \quad a \leftarrow a'$

    $V'_a \parallel V''_a \quad \Rightarrow \quad a =?$
  - *règles d'usages* : *e.g.*, réservations voyage non déficitaire $\rightsquigarrow a \leftarrow \max(a', a'')$
  - dépend de l'application, des utilisateurs...



Règles d'usages

- Limitations humaines
- Exemple : partage de données
- Exemple : Parker et perte connexité
- Exemple : travail hors connexion
  - Oracle 8i, commerciaux et portables
  - travail en parti connecté, en parti déconnecté
  - en cas de conflits, qui l'emporte ?
    - système : ancienneté
    - objet : meilleur client ? plus gros contrat ?
    - acteur : supérieur hiérarchique ?



Exemple d'applications

- Informatique nomade en milieu équipé
  - Mobile IP, IPv6
  - Cellulaire, GPRS, UMTS

  $\rightsquigarrow$ complexité / hostilité : proche d'Internet

Exemple d'applications

- Informatique nomade en milieu équipé
- Informatique en milieu non équipable
  - commandos + adhoc
  - secours dans des environnements dévastés...
  - régates...

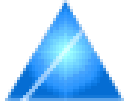
  ⇝ complexité ok, hostilité : coopération

Exemple d'applications

- Informatique nomade en milieu équipé
- Informatique en milieu non équipable
- **Informatique en milieu trop mouvant**
  - Bluetooth
  - exemples :
    - communication entre les équipements portés par l'utilisateur...
    - communications entre les objets de la maison

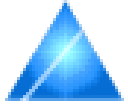
    ⇝ règles d'usages ?
  - Réunions publiques
    - échange de cartes de visites...
    - réception de la présentation sur les portables
    - échanges de messages entre deux personnes...

    ⇝ hostilité ?



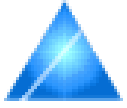
Exemple d'applications

- Informatique nomade en milieu équipé
- Informatique en milieu non équipable
- Informatique en milieu trop mouvant
- Excès ?
  - dans les services imaginés ? hum...
  - dans les technologies envisagées ?
  - complexité vs. règles d'usages
    - interface qui essaye de nous aider...
    - rejet de l'application (*e.g.*, WAP)
  - hostilité vs. coopération / arbitre
    - agents / intrus / virus



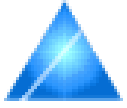
Exemple d'applications

- Informatique nomade en milieu équipé
- Informatique en milieu non équipable
- Informatique en milieu trop mouvant
- Excès ?
- Exemple d'application type
  - échange d'information sur le réseau routier sans infrastructure au sol
  - bornes fixes pour informations touristiques
  - multimédia : flashs, TV à bord, *etc.*
  - requêtes : recherche d'une information
  - *etc.*



Exemple d'applications

- Informatique nomade en milieu équipé
- Informatique en milieu non équipable
- Informatique en milieu trop mouvant
- Excès ?
- Exemple d'application type
  - échange d'information sur le réseau routier sans infrastructure au sol
  - arrivées / départs / mobilité
  - connexion / déconnexion
  - localisation relative / absolue
  - gestion du trafic induit, sélection meilleure source
  - *etc.*



Exemple d'applications

- Informatique nomade en milieu équipé
- Informatique en milieu non équipable
- Informatique en milieu trop mouvant
- Excès ?
- Exemple d'application type
  - échange d'information sur le réseau routier sans infrastructure au sol
  - complexité ?
    objectifs restreints, pas de partage de données
  - hostilité ?
    terminal sans données personnelles
  - service fourni sans mise en péril, en échange coopération raisonnable

Plan

- Définition
- Objectifs
- Modèles
  - mobiles, communications, réseau
  - implications, uniformité ? / “généricité” ?
  - où situer les développements ? démarche
- Techniques algorithmiques
- Conclusion

Modèle

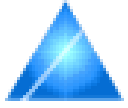
- Cadre pour l'étude
  - comment aborder un système réparti dynamique ? formaliser, prouver, comparer, étudier la complexité...
  - fixer ce qui peut l'être
  - diversité des réalités, et des paramètres
  - un "détail" peut tout changer. . .
  - modèle générique vs. spécifique
    usage large / réutilisation vs. précision / pertinence

Modèle

- Cadre pour l'étude
- En touche
  - caractéristiques des utilisateurs
    - perception de l'application : complexité règles, aides, interface, *etc.*
    - perception du risque : hostilité sécurisation, arbitre, coopération, *etc.*
  - caractéristiques des applications
    - capacité de coopération
    - ouverture vs. *overhead*
    - gestion du contexte
  - pas les technologies

Modèle

- Cadre pour l'étude
- En touche
- Paramètres élémentaires
  - Unité de mobilité
    - terminal, utilisateur, code, objets, variables. . .
    - unité de mobilité = unité d'exécution
  - Caractéristiques des mobiles
    - capacité de calcul, de mémoire, de batterie...
    - déconnexions... (durée, fréquence, volontaires ?)
  - Localisation
    - absence ? notion de mobilité ?
    - relative (*e.g.*, robots) ? absolue (*e.g.*, GPS) ?
    - référenciel interne ? externe ?



Modèle

- Cadre pour l'étude
- En touche
- Paramètres élémentaires
- Paramètres généraux
  - mode de communication
  - type de réseau
  - uniformité, "généricité"

Mode de communication

- Voisinage mouvant
  - connaissance incertaine car évolution
  - découverte incessante : qui est là ? n'est pas là ?
    - *roaming*, application nomade
    - terminal sans fil
    - *boot* d'une station dans un réseau fixe

    $\rightsquigarrow$ sous-entend diffusion locale dans le voisinage suspecté
  - technique de découverte
    - proactive : le fournisseur du service diffuse
    - réactive : le demandeur du service induit la diffusion
    - choix / efficacité en fonction des situations

Mode de communication

- Voisinage mouvant
- Communication sélective ?
  - Possibilité de sélectionner un voisin ?
    i.e., communication un vers tous ou un vers un ?
  - une fois le voisinage identifié, facile de diffuser un message avec un seul destinataire
    cf. bus Ethernet
  - au niveau du modèle ?
    influence l'algorithmique
    e.g., *broadcast*

Mode de communication

- Voisinage mouvant
- Communication sélective ?
- Collision ?
  - diffusion locale $\rightsquigarrow$ collisions ?
  - variété des hypothèses dans la littérature : avec détection, sans détection, si détection, de toute façon pas toujours, *etc.*
  - influence l'algorithmique : différence entre
    - ne rien recevoir
    - deux voisins au moins émettent en même temps

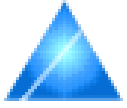
Mode de communication

- Voisinage mouvant
- Communication sélective ?
- Collision ?
- Quelle “généricité” ?
  - où situer les développements algorithmiques ?
  - grande diversité des modèles de communication

  ⇝ nécessite de clarifier, avec références aux technologies ; deux types de bibliographie.

Caractéristiques du réseau

- Réseau entièrement / partiellement dynamique
  - création / disparition / mobilité des nœuds
  - quels nœuds ?
  - sous-réseau fixe ?
  - *cf.* GSM vs. adhoc
  - influence l'application



Caractéristiques du réseau

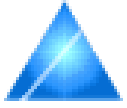
- Réseau entièrement / partiellement dynamique
- Exemple : sites fixes
  - un site fixe $\neq$ un site mobile à l'arrêt
    - plus sûr
    - plus de batterie
    - plus de puissance de calcul
    - plus de stockage
    - arbitre en cas d'hostilité

      (gestionnaire réseau fixe, hiérarchie...)
    - ...

  $\rightsquigarrow$ travail plutôt centralisé

Caractéristiques du réseau

- Réseau entièrement / partiellement dynamique
- Exemple : sites fixes
- Quelle uniformité ?

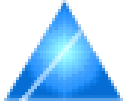


Où situer les dév. algorithmiques ?

- But
  - algorithmes adaptatifs aux déplacements et création / disparition des entités
  - quelles technologies ? choix assez large...
  - quels algorithmes ?
    - écueil : complexité / hostilité
    - application type : réseau routier

Où situer les dév. algorithmiques ?

- But
- Couches intermédiaires
  - pas plus bas que la couche liaison (MAC)
  - pas plus haut que la couche transport



Où situer les dév. algorithmiques ?

- But
- Couches intermédiaires
- Démarche
  - déterminer les aspects génériques / spécifiques algorithmiques plutôt que structurels
    $\rightsquigarrow$ modèle ?
    network computing model $\rightsquigarrow$ *abstract mobile model*
    masquer la répartition $\rightsquigarrow$ masquer la dynamique
  - identifier les primitives et services de base
    $\rightsquigarrow$ sorte de *middleware*...
    (localisation, évenements, QoS, adaptation...)
  - étudier des méthodes de développement
  - les mettre à l'épreuve sur des applications type

Plan

- Définition
- Objectifs
- Modèles
- Techniques algorithmiques
  - utilisation du routage
  - épine dorsale
  - point d'ancrage
  - agents
  - auto-stabilisation
  - détecteurs de défaillances
  - mobile $=$ message persistent
- Conclusion

Panorama des techniques algorithmiques

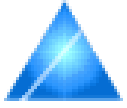
- Admettre l'existence d'un routage
  - le routage cache les évolutions du réseau
  - que reste-t-il à faire ?
    - adapter les algorithmes répartis classiques
    - optimiser en fonction de caractéristiques connues mobilité, dynamique, *overhead*, *etc.*

Panorama des techniques algorithmiques

- Admettre l'existence d'un routage
- Que faire sans le routage ?
  - routage = émuler un réseau logique complet sur un réseau physique qui ne l'est pas
  - le routage n'est pas toujours nécessaire
    - estampilles, instantanés
    - détection de terminaison
    - exploration de réseaux
    - construction du routage (!)
    - *etc.*

Panorama des techniques algorithmiques

- Admettre l'existence d'un routage
- Que faire sans le routage ?
- Techniques algorithmiques
  - épine dorsale
  - point d'ancrage
  - agents
  - auto-stabilisation
  - détecteurs de défaillances
  - mobile = message persistant
  - . . .



Exploitation du routage

- But
  - adaptation des algorithmes classiques au dessus de la couche réseau
  - optimisation des algorithmes
- Exemple : exclusion mutuelle [Baldoni02]
  - permission / jeton
  - demande de jeton ou anneau
  - ici mélange de demande jeton et circulation
  - anneau construit *on the fly* par le privilégié
  - politique = le plus proche en terme de nombre de sauts (fourni par le routage)

Utilisation d'une épine dorsale

- Colonne vertébrale dans l'application *backbone*
  - les mobiles s'y rattachent
  - les communications l'utilisent (*broadcast*)
  - entre routage et absence de routage
  - éventuellement différence entre les nœuds du *backbone* et les autres
    - caractéristiques physiques différentes
    - rôle différent (*e.g.*, arbitre)

Utilisation d'une épine dorsale

- Colonne vertébrale dans l'application *backbone*
- En pratique
  - réseaux avec partie fixe (*e.g.*, GSM)
  - réseaux sans partie fixe
    $\rightsquigarrow$ arbre couvrant mouvant

Utilisation d'une épine dorsale

- Colonne vertébrale dans l'application *backbone*
- En pratique
- Difficultés
  - croisement du mobile et du message à délivrer
  - coût du maintient face à la dynamique des mobiles

Point d’ancrage

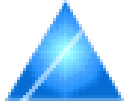
- Point d’ancrage vers un mobile
  - pointeur pour indirection vers le mobile
  - le mobile informe son point d’ancrage de sa nouvelle localisation
  - retransmission des messages (*forwarding*)

Point d’ancrage

- Point d’ancrage vers un mobile
- En pratique
  - Réseau avec partie fixe
    cf. home agent dans Mobile IP
  - Réseau sans partie fixe ?

Point d'ancrage

- Point d'ancrage vers un mobile
- En pratique
- Difficultés
  - perte de messages entre migration et mise à jour ; si retransmission, le mobile pourrait rebouger, *etc.*
  - nombreuses com. entre point d'ancrage et mobile ; phénomène accentué avec la réduction des cellules
  - *multicast*
    - abonnement *via* le point d'ancrage : les paquets *multicast* deviennent *unicast*
    - abonnement local dans le sous-réseau visité : disponibilité d'un routeur *multicast* local, conversions d'adresses



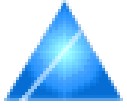
Agents mobiles

- Requêtes *offline*
  - exemple Oracle
    gestion à distance, réplication partielle
  - travail en mode déconnecté
  - requêtes par agent
- Parcours de réseaux
  - découverte / recherche
  - marches aléatoires
- Reproduction
  - clonage
  - création / disparition à distance

[Stefano02]

Auto-stabilisation

- Algorithme auto-stabilisant
  - supporte les pannes temporaires
  - panne = modification message ou variable
  - code préservé (ou restauré), *e.g.*, système câblé
  - l'algorithme retrouve un comportement correct après un temps fini



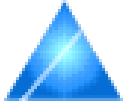
Auto-stabilisation

- Algorithme auto-stabilisant
- Intérêt
  - correction sans intervention humaine
  - plus d'initialisation
  - changement de l'environnement
    $\rightsquigarrow$ modification d'une variable

Auto-stabilisation

- Algorithme auto-stabilisant
- Intérêt
- Applications
  - adaptation aux changements du trafic...
  - changement dans le voisinage
    $\rightsquigarrow$ changement variable
  - déconnexion / reconnexion
  - apparition ?
  - disparition définitive ?

  $\rightsquigarrow$ tâches statiques / dynamiques...



Auto-stabilisation

- Algorithme auto-stabilisant
- Intérêt
- Applications
- Difficultés
  - différents modèles, hiérarchie
  - complexité des preuves
  - lecture de l'état des voisins (mémoire partagée)
  - lecture de ce qu'ils veulent bien laisser lire (registres)
  - passage de messages (réseaux)

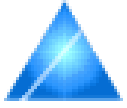
  $\rightsquigarrow$ cycle de développement des algorithmes
  intérêt des r-opérateurs [DT01]

Détecteur de défaillance

- Consensus
  - chaque entité correcte choisit une valeur
  - une valeur choisie doit avoir été proposée
  - pas de valeurs différentes
  - impossible en asynchrone si panne

Détecteur de défaillance

- Consensus
- Détecteur de défaillance non fiable
  - liste locale d'entités suspectées d'être en panne
  - propriétés de complétion
    e.g., une entité en panne finit par être suspectée continuellement
  - propriétés de précision
    e.g., une entité correcte finit par ne plus jamais être suspectée
  - détecteur de défaillances non fiables



Détecteur de défaillance

- Consensus
- Détecteur de défaillance non fiable
- Application à la mobilité
  - partie du réseau fixe [Seba02]
  - consensus entre les mobiles
  - reporter le problème sur les stations de base
  - mais lesquelles sont concernées ? (mobilité)
  - problème d'appartenance à un groupe dynamique (les stations de bases concernées)
    $\leadsto$ consensus
  - bilan :
    - deux problèmes de consensus en parallèle
    - mobilité $\leadsto$ détecteurs de défaillances

Détecteur de défaillance

- Consensus
- Détecteur de défaillance non fiable
- Application à la mobilité
- Extensions ?
  - absents $\rightsquigarrow$ défaillants
  - création $\rightsquigarrow$ entité correcte détectée
  - disparition d'entité $\rightsquigarrow$ défaillance détectée
  - réapparition d'entité $\rightsquigarrow$ fin de défaillance détectée

$\rightsquigarrow$ vers un cadre pour gérer la mobilité ?

Mobile = message persistant

- Unité de mobilité $\equiv$ message
  - réseaux type cellulaire (partie fixe) [Murphy00]
  - cellule = nœud
  - mobile = message persistant
  - autres messages : voix, *etc.*
  - handover = traversée d'un canal entre deux cellules (noeuds)

Mobile = message persistant

- Unité de mobilité $\equiv$ message
- Intérêt
  - modèle similaire aux modèles habituels
  - facilite l'application d'algorithmes répartis connus dans le contexte de la mobilité

Mobile = message persistant

- Unité de mobilité $\equiv$ message
- Intérêt
- Délivrance des messages
  - quand le mobile est sur un canal entre deux cellules, il peut être vu comme momentanément déconnecté
  - robustesse de la délivance des messages ?
    - graphe fortement connexe FIFO
    - communication sûre entre les nœuds (cellules)
    - un message à un nœud laissé par un mobile
    - un destinataire mobile ailleurs

Mobile = message persistant

- Unité de mobilité $\equiv$ message
- Intérêt
- Délivrance des messages
  - quand le mobile est sur un canal entre deux cellules, il peut être vu comme momentanément déconnecté
  - robustesse de la délivance des messages ?

  $\rightsquigarrow$ assurer que le message arrive une et une seule fois au mobile sans laisser de trace après un temps fini
  - borne sur le temps de stokage...

Diffusion *via* un instantané

- Instantané (*snapshot*)
  - état global cohérent, sites + canaux
  - Chandy Lamport, marqueur
  - Lai et Yang, lestage
  - gel de l’application

Diffusion *via* un instantané

- Instantané (*snapshot*)
- Utilisation
  - mobile = message, donc noté dans l'état global
    - utiliser l'état global pour délivrer le message au mobile ?
    - il pourrait bouger entre temps
  - modification de l'algorithme d'instantané
    - sauvegarde message-mobile
      $\rightsquigarrow$ délivrance du vrai-message
    - pas de collecte de l'état du système

Diffusion *via* un instantané

- Instantané (*snapshot*)
- Utilisation
- Algorithme de Chandy Lamport
  - FIFO
  - initiateur $\leadsto$ marqueur sur tous les canaux
  - première réception marqueur
    - sauvegarde locale
    - retransmission marqueur sur tous les canaux

Adaptation de l'instantané

- Modifications de l'algorithme
  - com. unidirectionnelle $\rightsquigarrow$ graphe orienté symétrique
  - marqueur = message à délivrer (vrai-message)
  - message = mobile (message-mobile)
  - arrivée marqueur $\rightsquigarrow$ arrivée du vrai-message
    - première réception :
      retransmission sur les canaux de sortie
    - message-mobile présent :
      il reçoit le vrai-message
    - sinon :
      le nœud-cellule conserve le message jusqu'à recevoir le marqueur par tous les canaux entrants
  - arrivée du message-mobile sur un nœud averti par un canal non encore visité $\rightsquigarrow$ délivrance du message

Adaptation de l'instantané

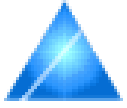
- Modifications de l'algorithme
- Preuve
  - pas de stockage résiduel
    - connexité $\rightsquigarrow$ un message par canal
    - après nettoyage, pas d'arrivée de marqueur possible
  - chaque message est délivré au destinataire
    - mobile sur un nœud
    - mobile sur un canal visité par le marqueur
  - chaque message n'est délivré qu'une fois
    - mobile sur un nœud qui reçoit le marqueur pour la première fois
    - mobile arrivant par un canal non visité sur un nœud averti

Hypothèse des canaux FIFO

- Canaux FIFO ?
  - un seul canal entre deux cellules ?
  - les messages sont plus rapides que les mobiles (!)

Hypothèse des canaux FIFO

- Canaux FIFO ?
- Handover
  - dégradation du signal
  - la station de base A cherche une station voisine B
  - requête sur les fréquences entre les stations de base
  - la station A indique la nouvelle fréquence au mobile (message `switch`)
  - le mobile envoie un “hello” à la station B
  - la station B avertit la station A que le handover est fini



Hypothèse des canaux FIFO

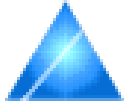
- Canaux FIFO ?
- Handover
- Problème
  - canaux FIFO entre stations de base pas suffisant
  - déséquencement entre
    - partie filaire (vrai message)
    - partie non filaire (`switch`)

Hypothèse des canaux FIFO

- Canaux FIFO ?
- Handover
- Problème
- Solution
  - message spécifique `VMU` envoyé par A à B, signifie que le mobile quitte la cellule A pour la B
  - `VMU` et `switch+hello` atomiques
  - avant `VMU`, les messages sont délivrés par A
  - après `VMU`, les messages sont délivrés par B
  - si le `hello` arrive avant le `VMU` en B, le mobile est considéré absent
  - si le `VMU` arrive avant le `hello` en B, les messages sont stockés jusqu'à l'arrivée du `hello`

Conclusion

- Algorithmique répartie adaptative
  - adaptation à la dynamique et la mobilité
  - des entités calculantes
- Objectifs
  - masquer la dynamique et la mobilité
  - développements à un niveau intermédiaire
  - briques de base
- Difficultés
  - vers un modèle réparti adapté ?
  - problèmes algorithmiques ardus
  - techniques algorithmiques ? (8 identifiées)
  - performances, convergences ?



Références

[Baldoni02] Baldoni, R., Virgillito, A. et R.Petrassi. *A distributed mutual exclusion algorithm for mobile ad-hoc networks* Proc. ISCC 2002 pp 539-544.

[DT01] Ducourthial, B. et Tixeuil, S. *Self-stabilization with r-operators*, Distributed Computing Juil. 2001.

[Murphy00] Murphy, A.L. *Enabling the rapid development of dependable applications in the mobile environment*. Thèse de doctorat, Université de Washington, Août 2000.

[Seba02] Seba, H., Badache, N. et Bouabdallah, A. *Solving the consensus problem in a dynamic group: an approach suitable for a mobile environment*

[Stefano02] Di Stefano, A. et Santoro, C. *Locating Mobile Agent in a Wide Distributed Environment*, IEEE TPDS, 13-8, août 2002 pp. 844-864.