Soutenance de thèse de

Quentin Monnet

Directrice :

Prof. Lynda Mokdad

Modèles et mécanismes pour la protection contre les attaques par déni de service dans les réseaux de capteurs sans fil

Vendredi 17 juillet 2015

Université Paris-Est

Présentation

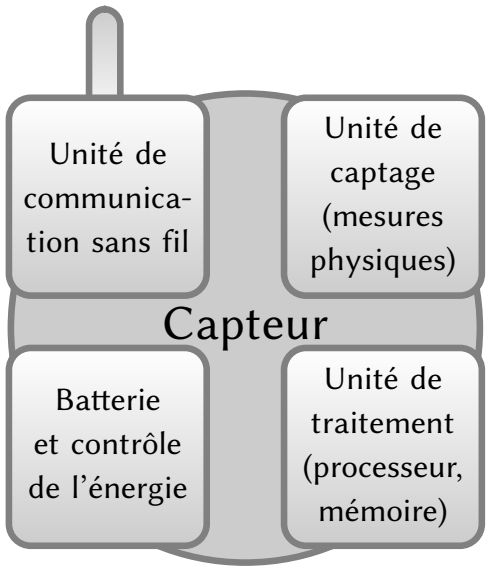
Capteurs

Capteurs (ou *nœuds*) (anglais : *sensors, nodes, motes*) : de petits appareils

- effectuant des mesures (lumière, CO_2, température, champ magnétique, vibrations, …)
- communiquant sans fil (*ad-hoc*)
- reliés à une station de base

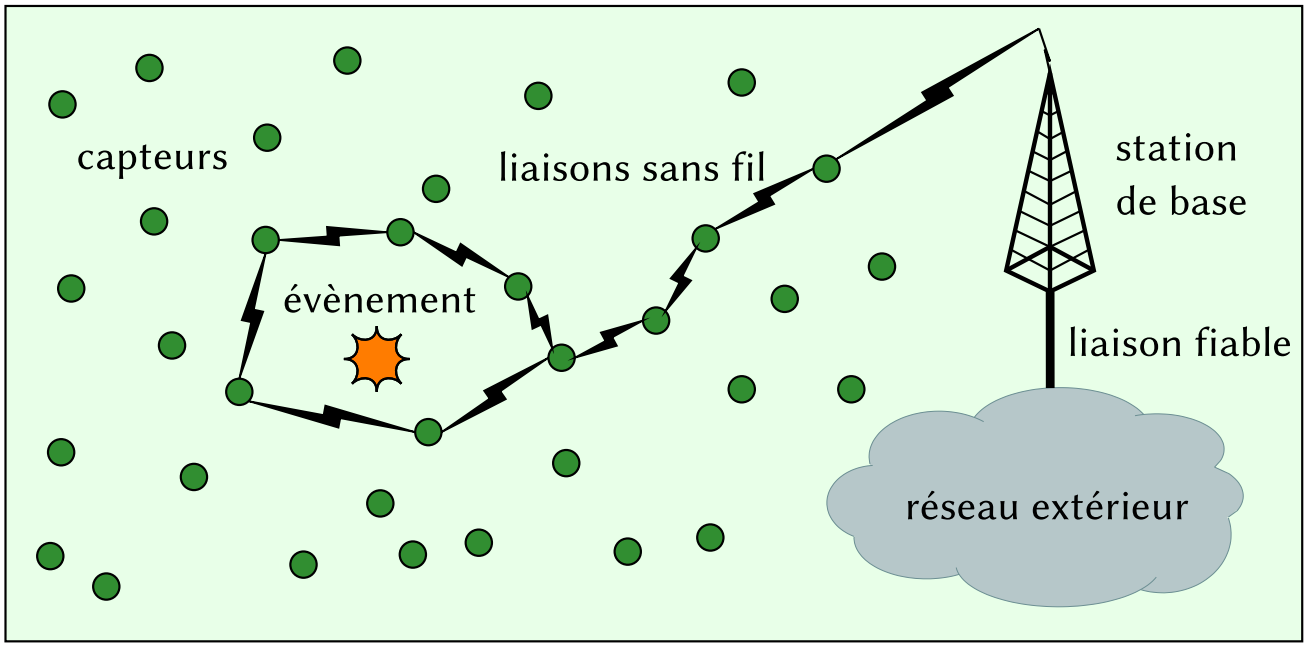
Ressources limitées

- Faibles capacités de calcul
- Peu de mémoire disponible
- Énergie limitée (batterie)



Réseaux de capteurs sans fils

Capteurs en réseau : *Wireless Sensor Networks (WSNs)*

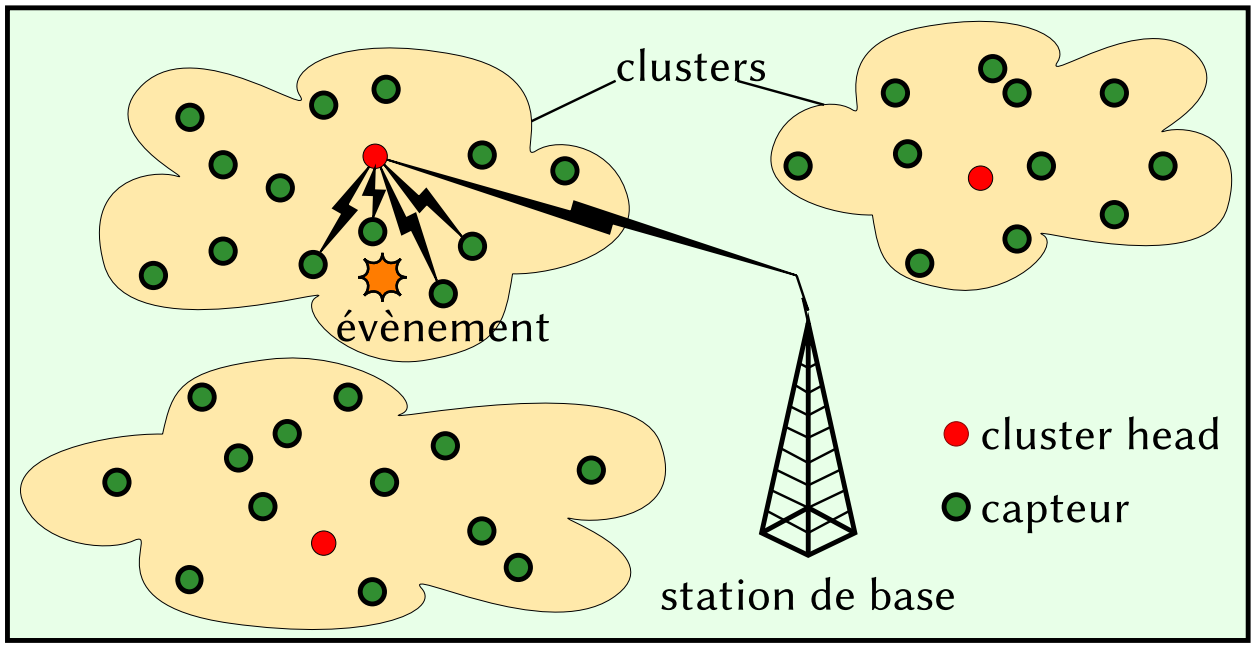


Réseaux de capteurs sans fils clusterisés

Quelques problématiques :

- déploiement autonome ; gestion décentralisée
- performances ; gestion de l'énergie
- sureté, résilience ; sécurité

Notion de clusters :



Exemples d'applications

- **Milieu urbain :** surveillance du trafic routier, « villes intelligentes »
- **Environnement :** agriculture, suivi d'animaux, mesure du taux de pollution, météo
- **Surveillance :** activité sismique, départs d'incendie en forêt ou sur site industriel, vidéosurveillance et détection d'intrusions (physiques)
- **Particuliers :** « Internet des objets », domotique
- **Médecine :** surveillance d'organes vitaux ou de glycémie, détection de tumeurs
- **Domaine militaire :** renseignement, détection d'agents chimiques / biologiques / radioactifs

Fortes contraintes en sécurité

Sécurité dans les réseaux de capteurs

Plusieurs composantes :

- Confidentialité des données
- Authentification, intégrité des échanges
- Disponibilité des services
  Dans notre cas : détection des attaques par déni de service (DoS).
  Plusieurs couches (pile TCP/IP) peuvent être visées :
  - couche physique (brouillage)
  - couche MAC (brouillage, comportements égoïstes, privation de sommeil, …)
  - routage (trous noirs, trous de vers, attaques sur le protocole de routage, …)
  - couche transport (tempêtes SYN/ACK sur TCP/UDP, …)
  - applications

Attaques étudiées

Objectif : détecter des capteurs compromis qui tenteraient de nuire au réseau depuis l'intérieur.

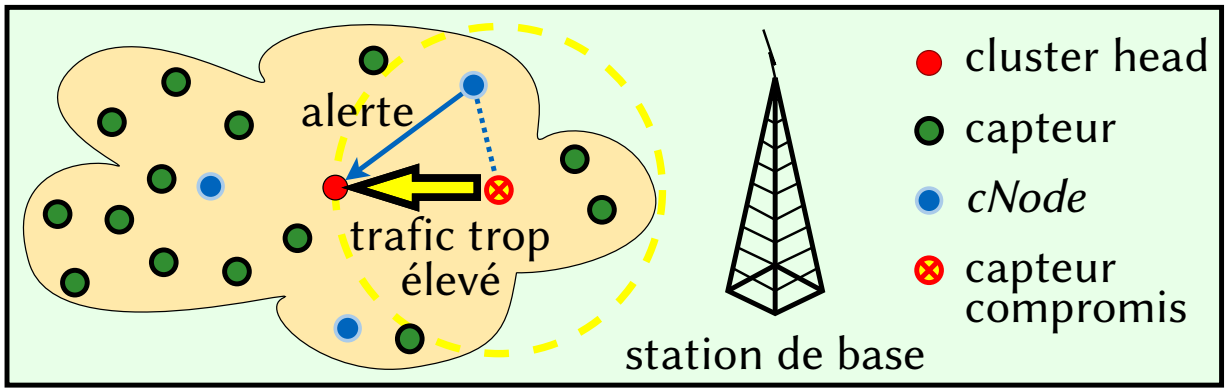
- Les capteurs compromis font partie du réseau (même matériel)
- Ils effectuent des attaques sur les couches MAC (contrôle d'accès au médium) ou de routage IP, par exemple :
  - non retransmission de paquets
  - brouillage, saturation du canal, comportement égoïste

Contrainte : limiter et répartir la consommation en énergie

Détection des attaques

Mécanisme de détection des comportements suspects

- Sous-ensemble de capteurs (*cNodes*) chargés d'établir une surveillance du trafic
- Application d'un ensemble de règles sur le trafic observé
- Si détection d'un comportement suspect, remontée d'une alarme au cluster head



- Renouvellement périodique des *cNodes*

Problème posé :

Pour chaque période, comment sélectionner les *cNodes* ?

Trois méthodes de renouvellement

1. Sélection aléatoire
2. Sélection selon l'énergie résiduelle
3. Élection démocratique

Résultats numériques : simulations

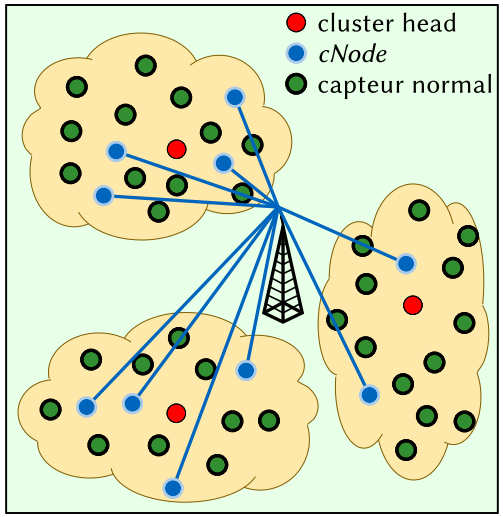
Sélection aléatoire des *cNodes*

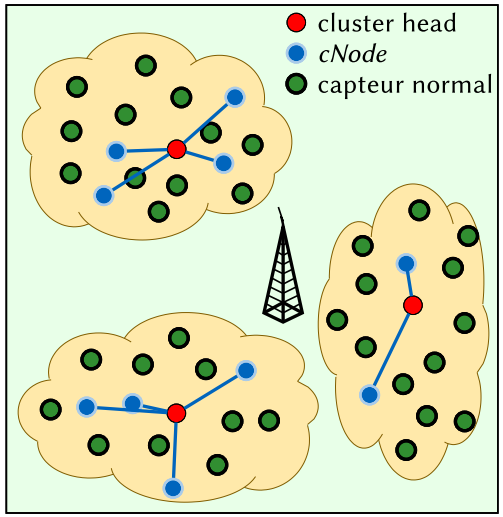
- Méthode simple
- Dans la littérature : rien sur le renouvellement de la sélection des *cNodes* au cours du temps [Lai et Chen, 2008]
- Ce qui importe : assurer ce renouvellement

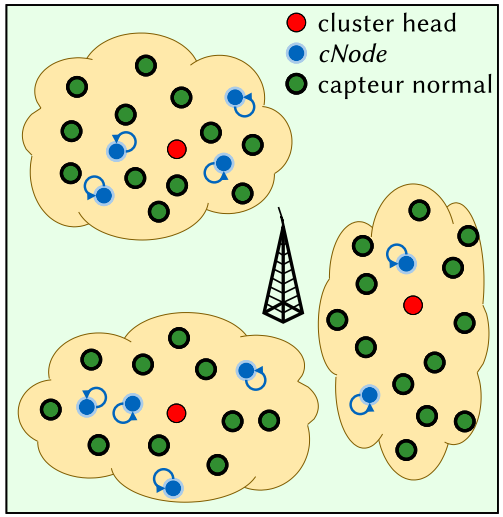
Principe

Déterminer la liste des *cNodes* de façon aléatoire, à l'aide d'un générateur de nombres (pseudo-)aléatoires

Sélection aléatoire des *cNodes* — implémentation







Méthode	Avantages	Inconvénients
Sélection par la station de base	• Aucun calcul des capteurs • Distribution spatiale idéale	• Si le CH est compromis, il déclare un cluster vide • Perte de l'aspect décentralisé de l'algorithme
Sélection par les cluster heads	• Seuls les CH calculent les nombres aléatoires	• Si le CH est compromis, il ne désigne aucun *cNode*
Auto-sélection	• Très simple • Peu de données de contrôle	• Chaque capteur calcule un nombre aléatoire • Ignore la topologie du réseau

Sélection des *cNodes* selon l'énergie résiduelle

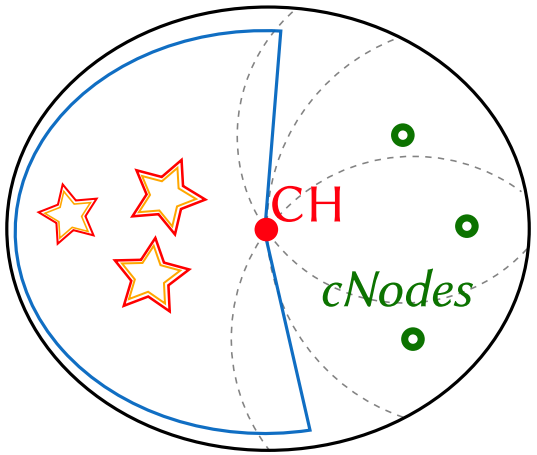
Principe

Les capteurs dont le niveau de charge de batterie est le plus élevé sont sélectionnés en tant que *cNodes*

Algorithme déterministe

Problèmes :

1. Comment empêcher les capteurs compromis d'accaparer le rôle de *cNode* ?
2. Comment être sûr de couvrir tout le cluster ?

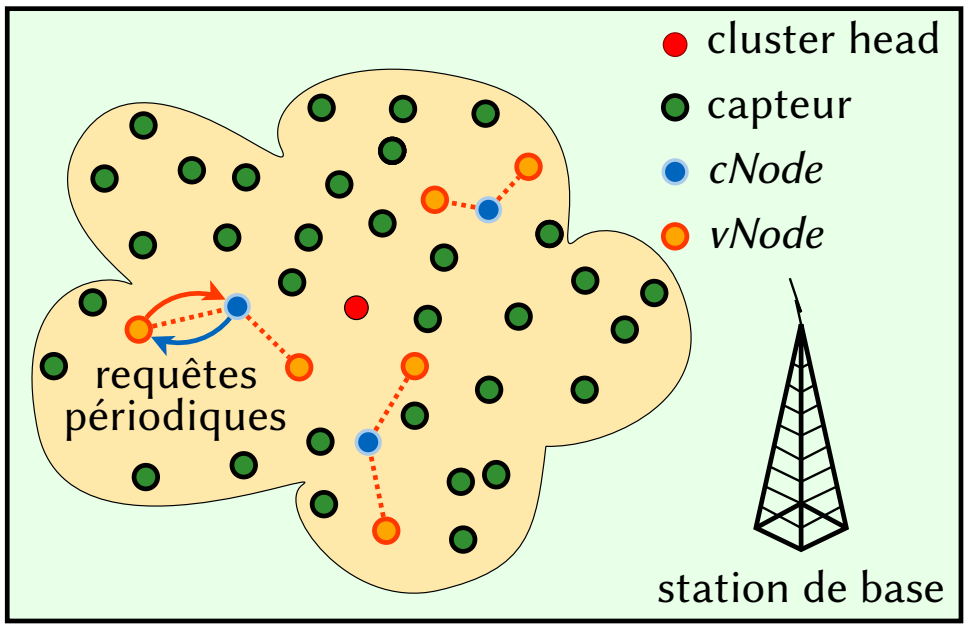


Sélection selon l'énergie résiduelle — solutions

Solutions :

1. Nouveau type de capteurs : les *vNodes*
2. Chaque capteur surveillé par au moins deux *cNodes*

Les *vNodes* surveillent la consommation énergétique des *cNodes* à l'aide d'un modèle mathématique



Élection démocratique des *cNodes*

Principe

Les observations réalisées par les capteurs sur leurs voisins sont utilisées pour élire (auprès du cluster head) les nouveaux *cNodes*

Deux étapes :

1. Une phase initiale où tous les capteurs observent leurs voisins
2. Puis fonctionnement standard : les *cNodes* envoient leurs observations (leurs « votes ») au cluster head

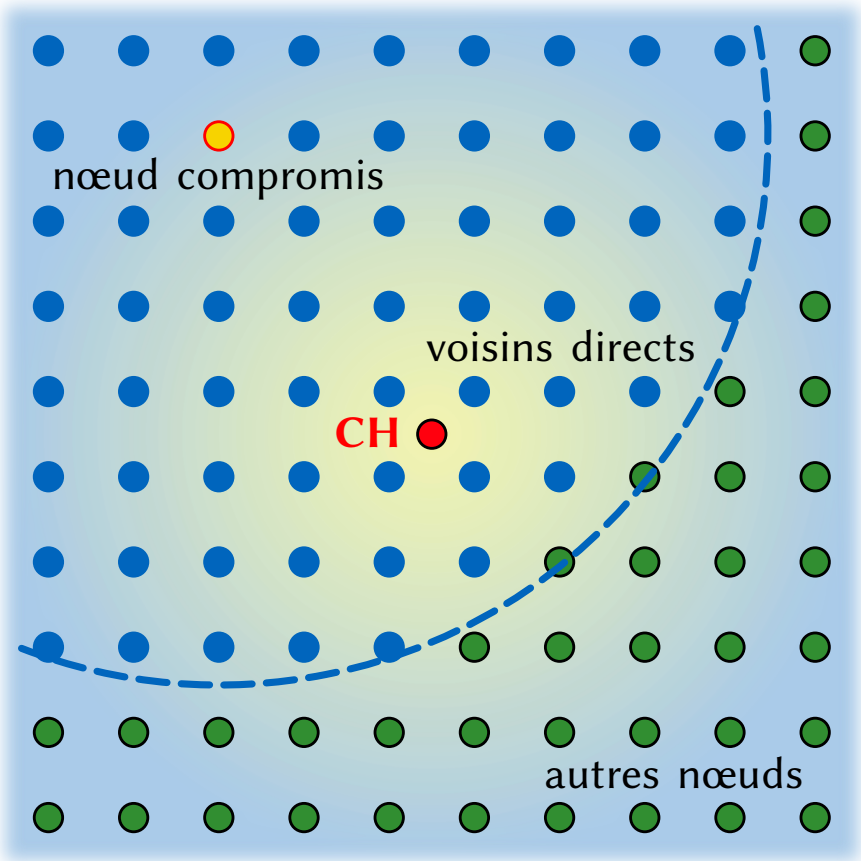
Le vote peut prendre en compte plusieurs critères :

$$\begin{aligned} note_k[i] &= (\alpha \times \textit{énergie\_résiduelle}_k[i]) + (\beta \times \textit{réputation}_k[i]) \\ &+ (\gamma \times \textit{index\_connectivité}_k[i]) + (\delta \times \textit{puissance du signal}_k[i]) \\ &+ (\zeta \times \textit{durée\_depuis\_dernière\_sélection}_k[i]) \end{aligned}$$

Simulations — système simulé

Logiciel utilisé : ns (network simulator)

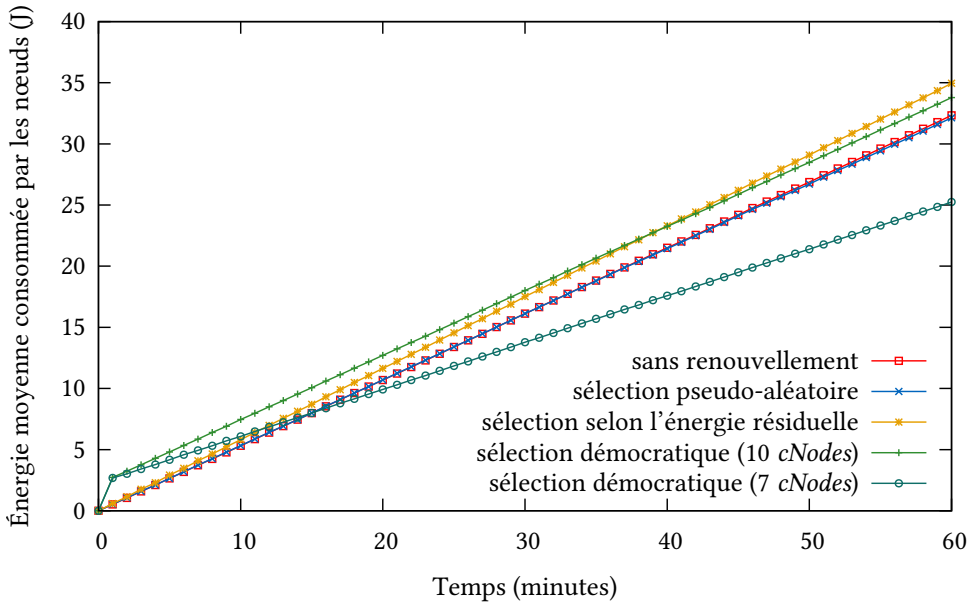
Grille de 100 capteurs



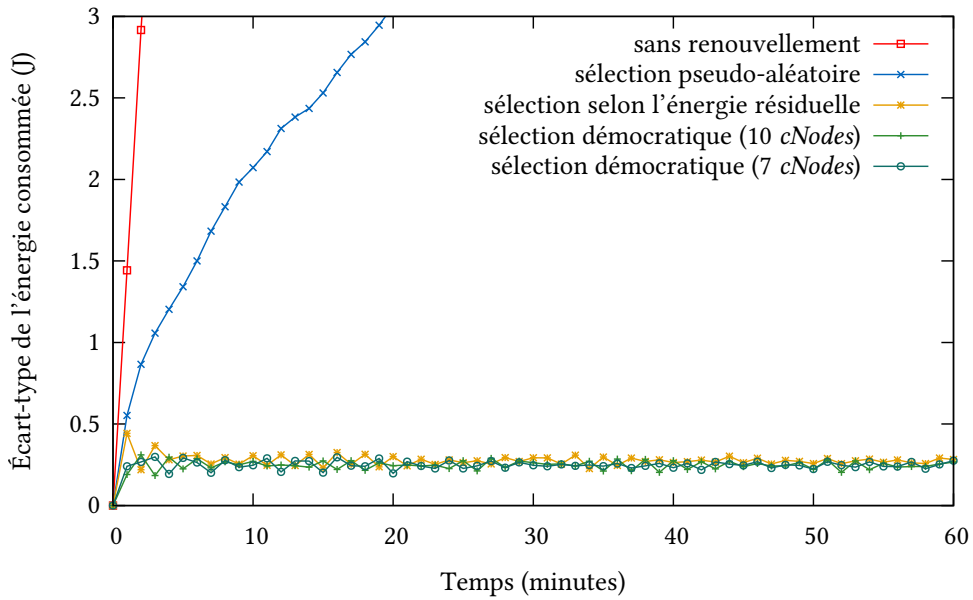
5 cas pour le renouvellement des *cNodes* :

- sans renouvellement
- aléatoire (10 *cNodes*)
- énergie résiduelle (10 *cNodes*)
- élec. dém. (10 *cNodes*)
- élec. dém. (7 *cNodes*)

Simulations — énergie consommée



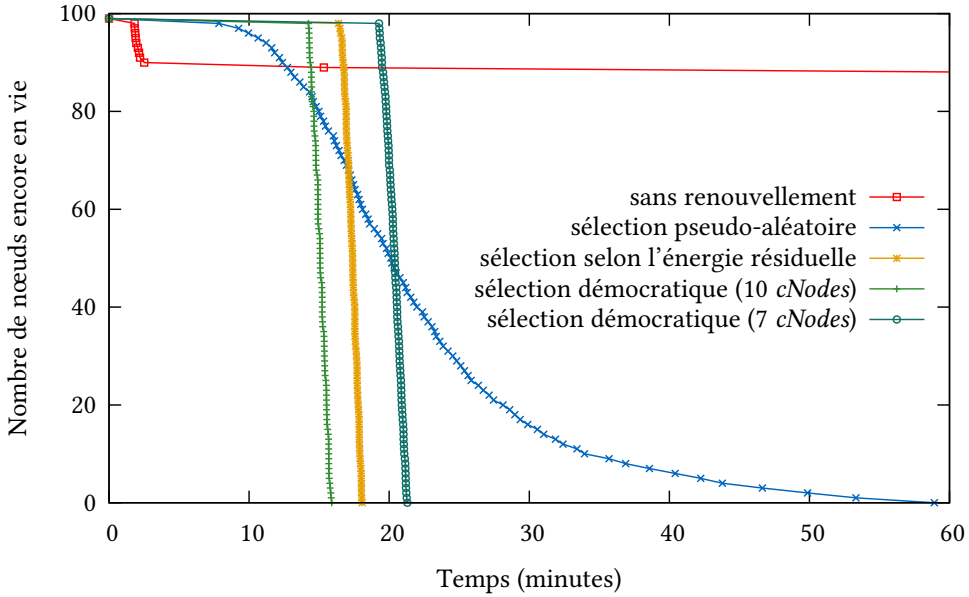
Énergie consommée (moyenne)

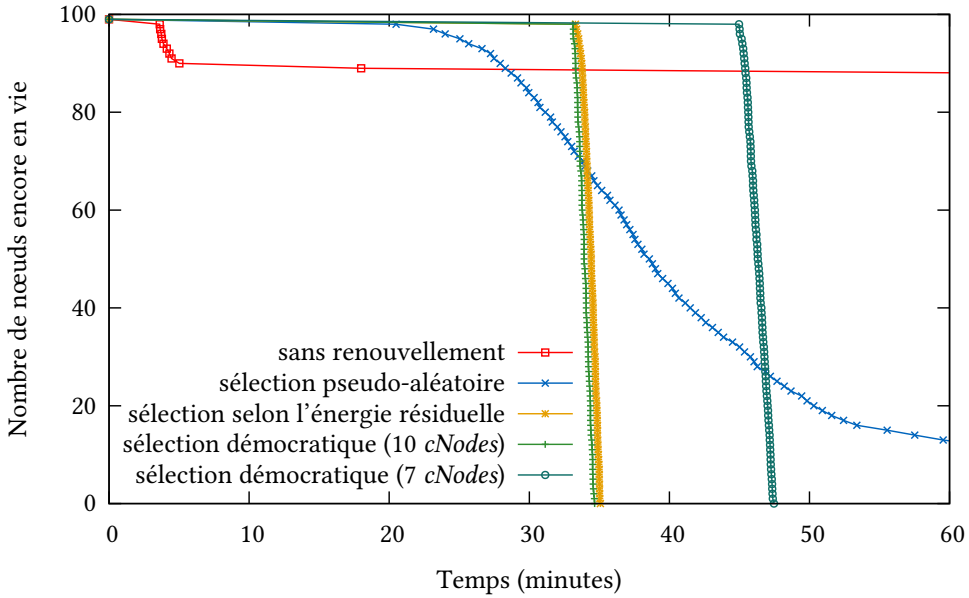


Énergie consommée (écart-type)

Simulations — nombre de capteurs « en vie »

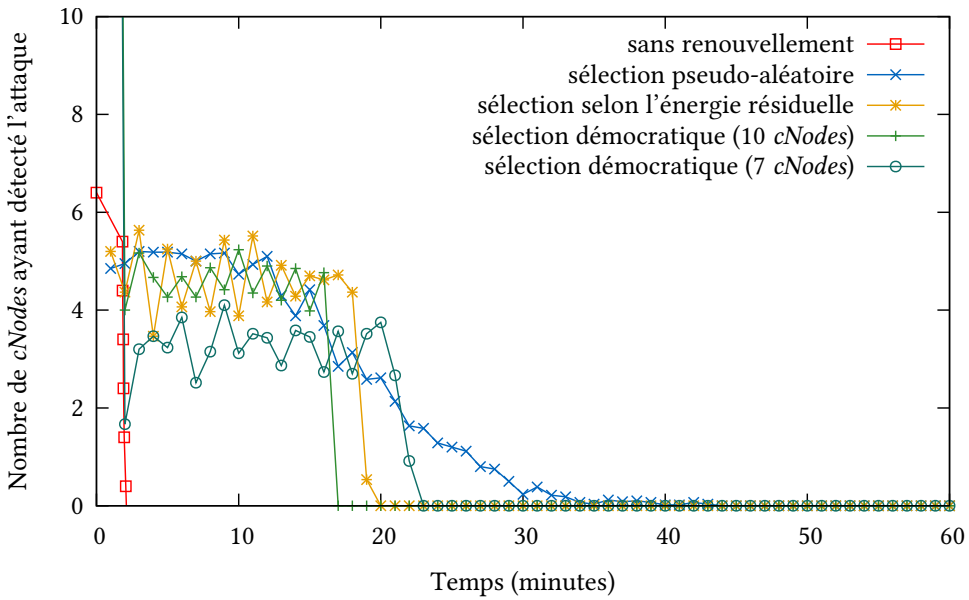
Nombre de capteurs en fonctionnement au cours du temps (énergie initiale : 10 J)





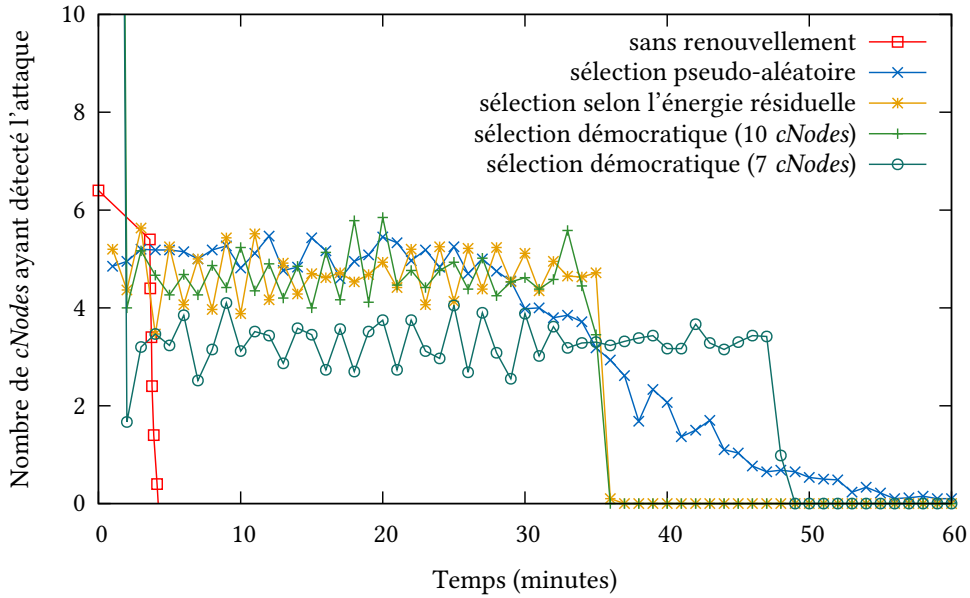
Nombre de capteurs en fonctionnement au cours du temps (énergie initiale : 20 J)

Simulations — détection au cours du temps



Nombre de *cNodes* détectant l'attaque au cours du temps (énergie initiale : 10 J)

Nombre de *cNodes* détectant l'attaque au cours du temps (énergie initiale : 20 J)



Avantages et inconvénients

Méthode	Avantages	Inconvénients
Aucun renouvellement	• Préservation des nœuds non sélectionnés	• Mauvaise surveillance
Sélection aléatoire	• Consommation modérée • Simple à mettre en œuvre • Pourcentage constant de *cNodes* • Bonne rotation des *cNodes* ; processus aléatoire : pas d'attaques	• Équilibre moyen de la charge • Risque : capteurs non couverts par les *cNodes* sur certaines phases
Sélection selon l'énergie résiduelle	• Bon équilibre de la charge • Surveillance de tous les capteurs par au moins deux *cNodes*	• *vNodes* contraignants ; couteux en énergie et lourds à implémenter • Très gourmand en énergie
Élection démocratique	• Bon équilibre de la charge • Surveillance de tous les capteurs par ≥ 2 *cNodes* • Consommation modérée (après période initiale) • Peut prendre en compte d'autres paramètres	• La période initiale consomme beaucoup d'énergie

Recommandations

Aucun renouvellement

Déconseillé, sauf si matériel dédié

Sélection aléatoire

Privilégie la longévité à la sécurité ; perte de certains capteurs plus tôt que d'autres

Sélection selon l'énergie résiduelle

Préférer l'élection démocratique ; sauf en cas de courtes périodes d'activité du cluster

Élection démocratique

Privilégie la sécurité ; maintien aussi longtemps que possible de l'intégralité des capteurs en fonctionnement

Différents modèles

Processus de détection et sélection aléatoire

1. Réseaux de Petri (RPGSe)
2. Logique stochastique (LSAH)

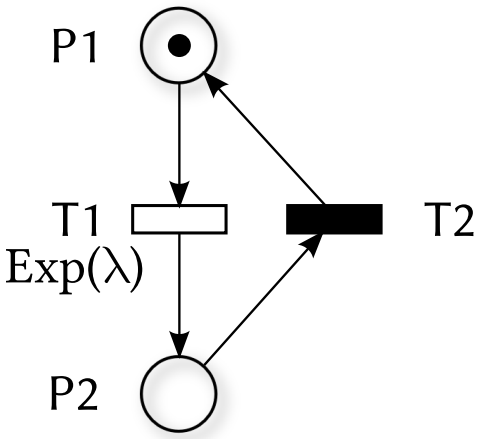
Interactions entre *cNodes* et capteurs compromis

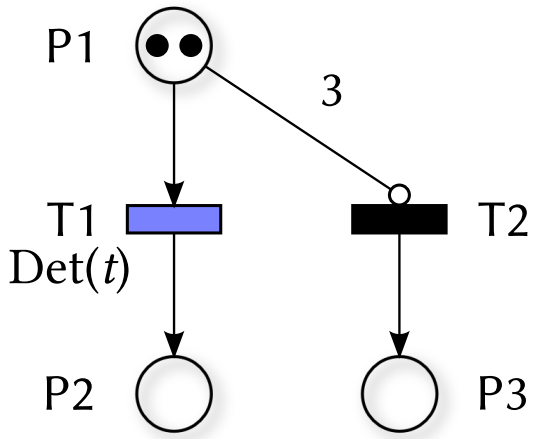
3. Jeux quantitatifs

Réseaux de Petri

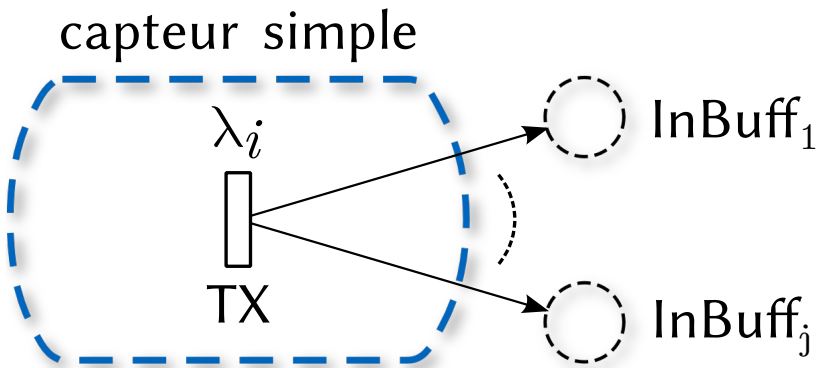
Réseaux de Petri stochastiques généralisés étendus (RPSGe)

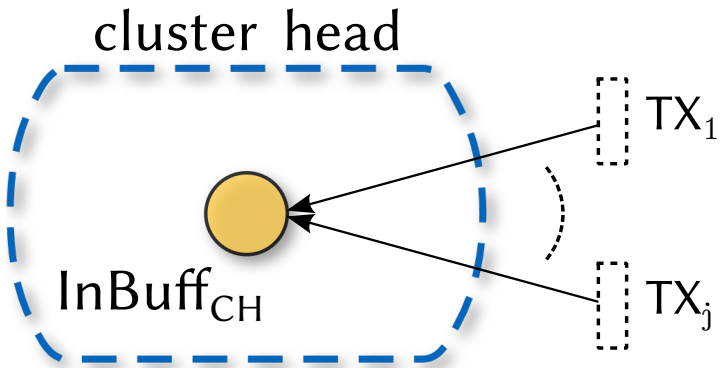
- utilisés pour modéliser des processus stochastiques
- transitions immédiates ou minutées (distribuées de façon exponentielle ou déterministe)
- arcs inhibiteurs

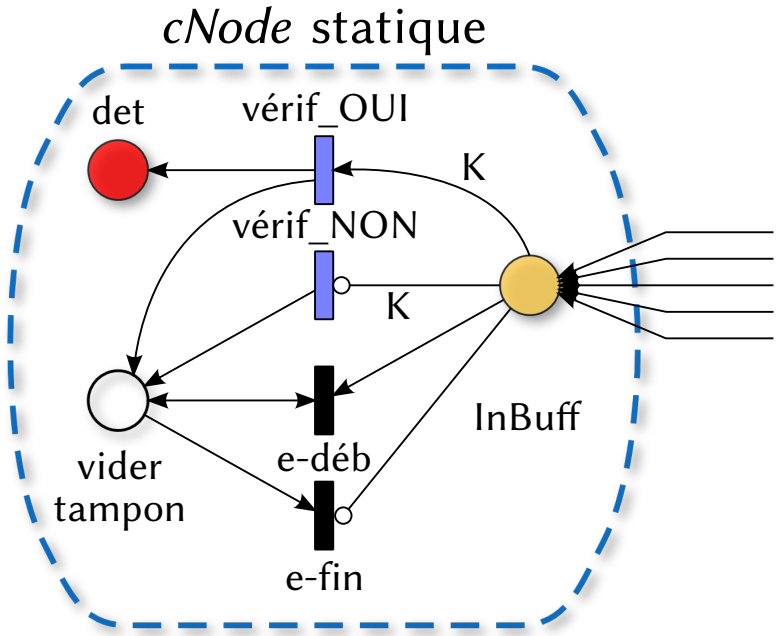




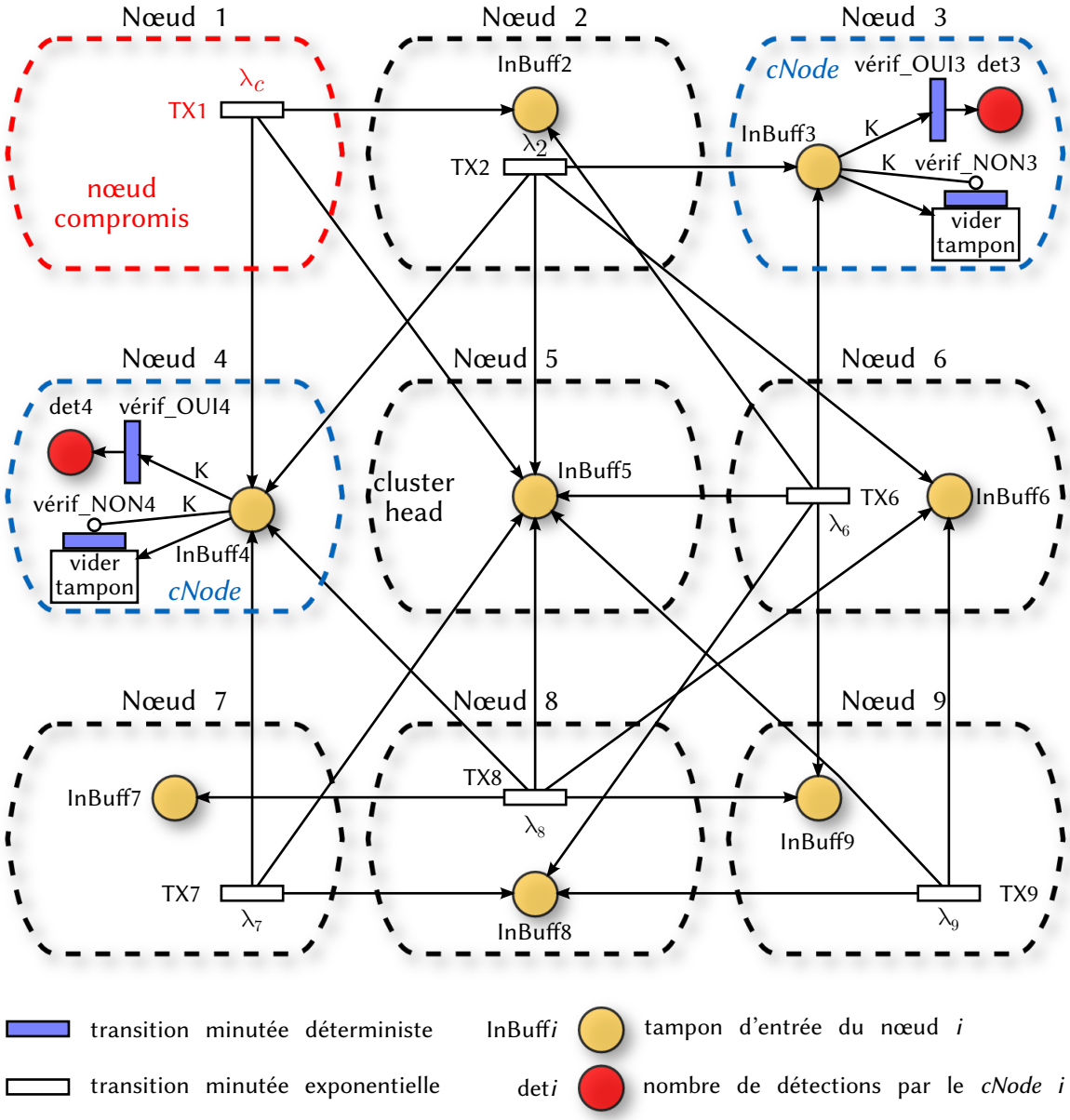
RPGSe : briques de base



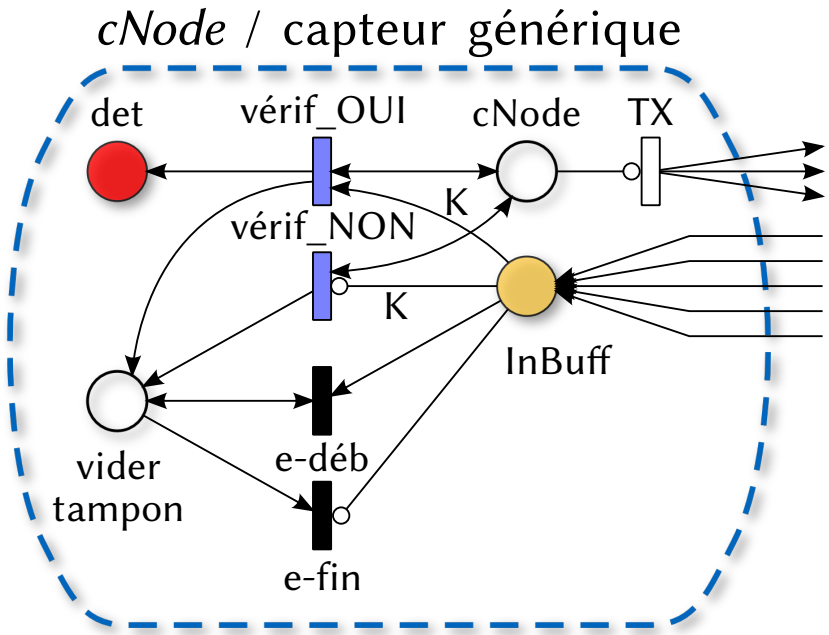




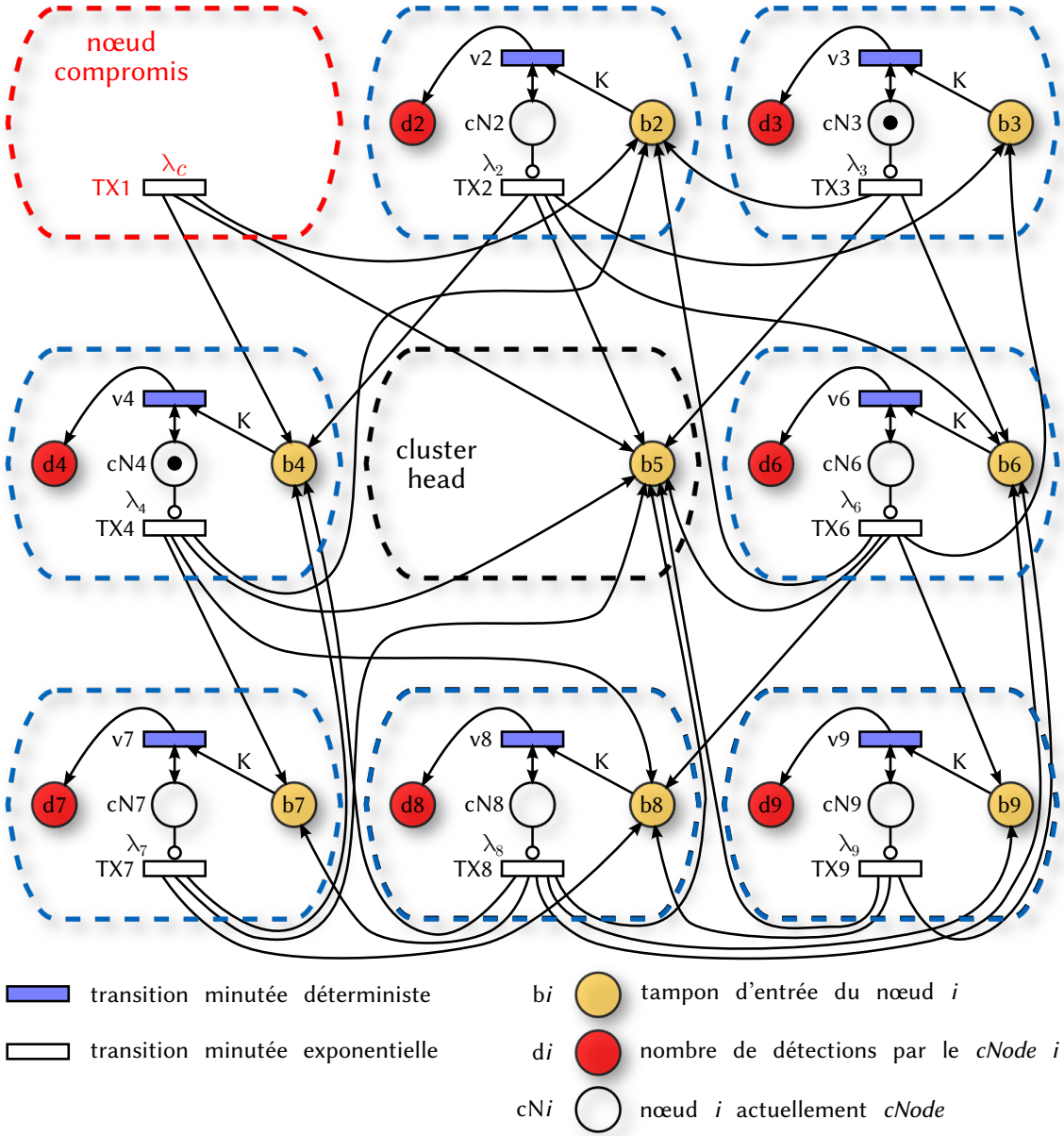
RPGSe : Réseau sans renouvellement des *cNodes*



RPGSe : Capteurs complets

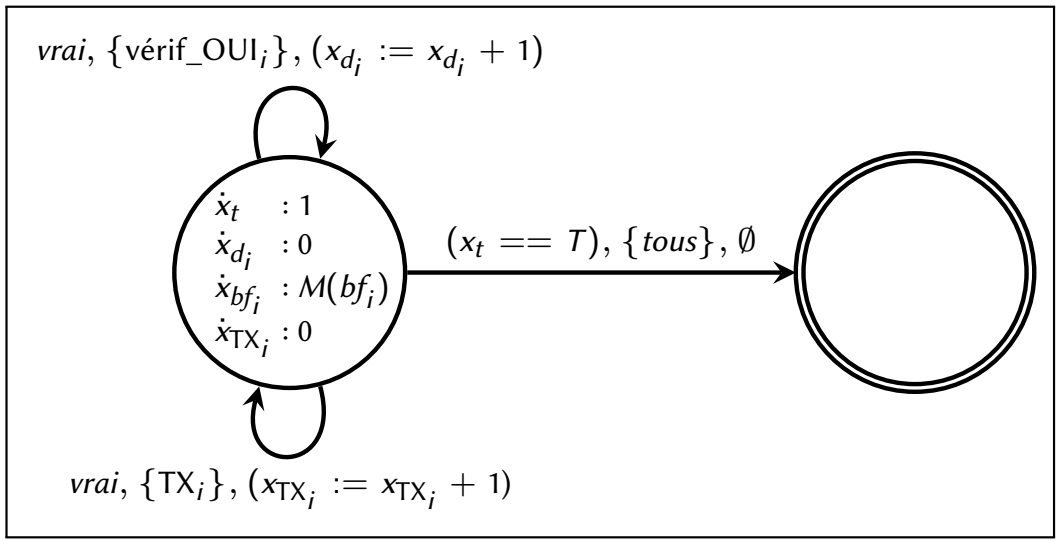


RPGSe : Réseau avec renouvellement des *cNodes*



Logique stochastique avec automates hybrides

- Basée sur un modèle RPSGe
- Mesures de performances exprimées en logique stochastique
- Une formule de cette logique comprend :
  - un automate linéaire hybride (ALH)
  - une expression construite à partir des variables de l'ALH
- Outils de *model checking* (COSMOS [Ballarini *et al.*, 2011]) pour vérifier ces propriétés



Exemples d'expressions :

$$Z_1 \equiv E(\text{dern}(x_{d_i}))$$
$$Z_2 \equiv E(\text{dern}(x_{d_i} + x_{d_{i'}}))$$
$$Z_3 \equiv E(\text{dern}(x_{TX_i}))$$
$$Z_4 \equiv E(\text{int}(x_{bf_i}))$$

Jeux quantitatifs

Principe :

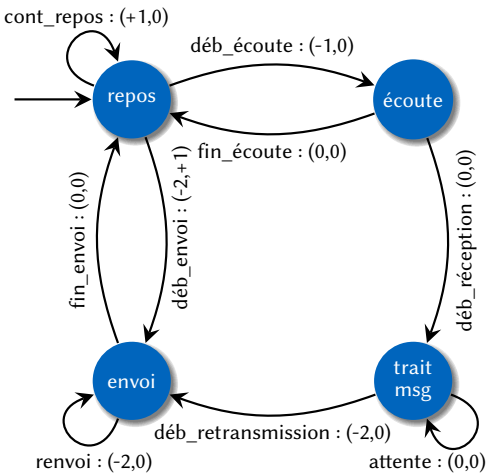
- Formule de gain à vérifier pour obtenir la victoire
- Problème de victoire : pour une configuration initiale et une formule de gain, existe-t-il une stratégie permettant d'obtenir la victoire ?
- Problème du crédit initial : existe-t-il une valeur pour le crédit initial pour laquelle le problème de victoire a une réponse positive ?

Plusieurs composantes pour les formules de gain : énergie et « gain » (messages envoyés avec succès)

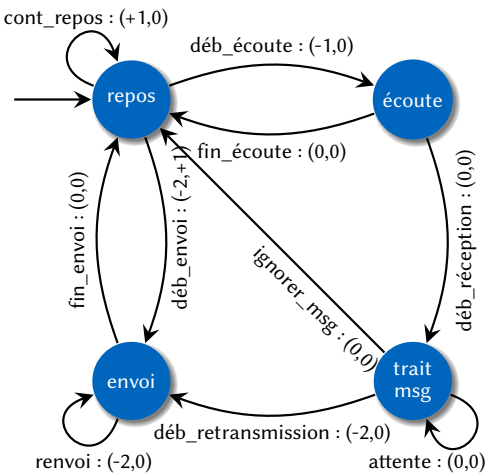
Jeux quantitatifs — graphes

Résultats :

- Cas général indécidable
- Conjonction d'atomes : possibilité de calculer une stratégie, potentiellement à mémoire infinie
- Dans ce dernier cas, possibilité de déterminer une approximation avec mémoire finie



Capteur légitime



Capteur compromis

Conclusion et perspectives

Conclusions

- Détection : trois mécanismes de renouvellement des *cNodes*
- Les résultats numériques indiquent une bonne répartition de la consommation
- Différents modèles pour représenter ces outils, en déduire des propriétés

Conclusion et perspectives

Travaux futurs

- Poursuivre l'étude du modèle de jeux quantitatifs
- Analyser les systèmes obtenus à l'aide d'outils de *model-checking*
- Varier les modèles et les solutions, rechercher d'autres méthodes de renouvellement ou même de détection
- Confronter les mécanismes proposés à des applications réelles (plate-forme opérationnelle)

Conclusion et perspectives

Travaux futurs

- Poursuivre l'étude du modèle de jeux quantitatifs
- Analyser les systèmes obtenus à l'aide d'outils de *model-checking*
- Varier les modèles et les solutions, rechercher d'autres méthodes de renouvellement ou même de détection
- Confronter les mécanismes proposés à des applications réelles (plate-forme opérationnelle)

Merci beaucoup !

Questions