



Modélisations et analyses de réseaux de capteurs

Ludovic Samper

Soutenance de thèse, le 7 avril 2008

France Télécom R&D / VERIMAG

JURY

Marc Pouzet

Isabelle Guérin-Lassous

Robert de Simone

Dominique Barthel

Florence Maraninchi

Laurent Mounier

LRI, Université Paris-Sud

LIP, Université Lyon I

INRIA Sophia-Antipolis

France Télécom R&D

Verimag, GrenobleINP

Verimag, Université Joseph Fourier

Président

Rapporteuse

Rapporteur

Examineur

Directrice de thèse

Directeur de thèse

Contexte

Thèse **CIFRE** France Télécom R&D / Verimag

Projet RNRT **ARESA** (2006-2009) sur les réseaux de capteurs, partenaires et compétences :

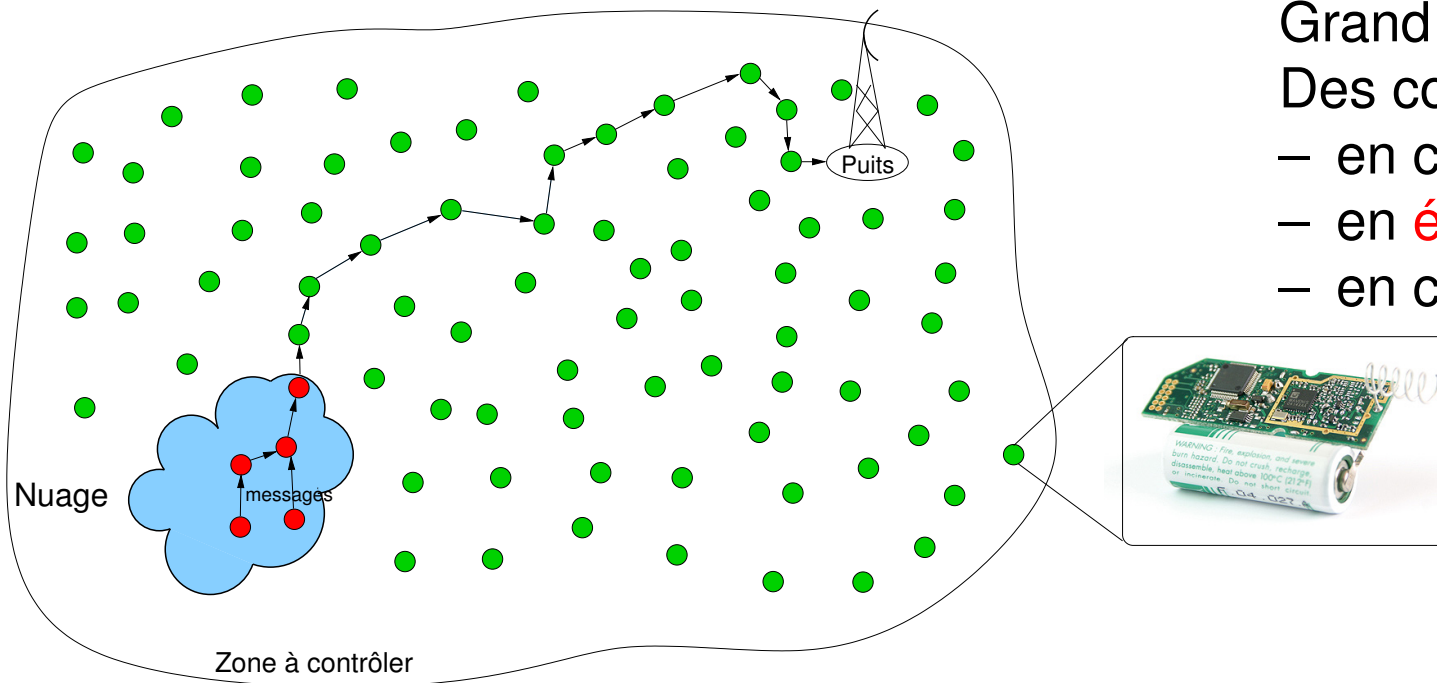
- Citi, auto-organisation
- LIG, protocoles MAC et routage
- Tima, architecture asynchrone
- **Coronis Systems, déploie des réseaux** ultra-basse consommation
- Verimag, modélisation
- France Télécom R&D, coordinateur du projet

Les réseaux de capteurs

Les capteurs :

- Mesure d'une grandeur physique
- Traitement des données
- Communication sans fil

Exemple d'application : détection d'un nuage toxique



Grand réseau

Des contraintes fortes :

- en coût
- en **énergie**
- en capacité de calcul

La recherche et les réseaux de capteurs

- Programmation des nœuds
- Protocoles de communication
- Matériel
- Sécurité
- **Modélisation**

Importance de la modélisation

Difficultés pour concevoir un réseau de capteurs **efficace en énergie** :

- Programmation des nœuds,
- Choix des protocoles de communication (et de leurs paramètres),
- Optimisations croisées (e.g. MAC/routage),
- Choix du matériel

Conception d'une **solution optimale** probablement hors de portée

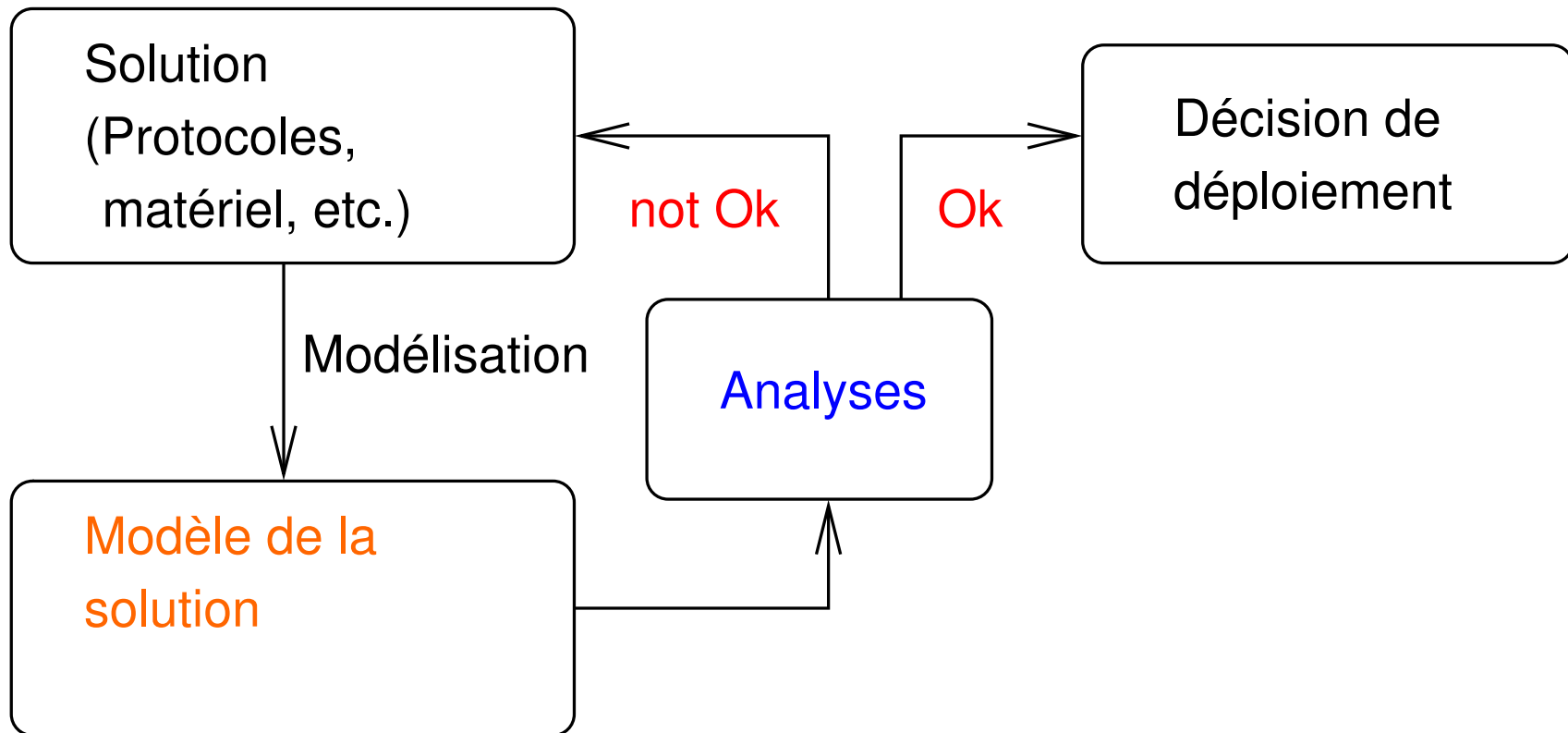
On voudrait simplement vérifier qu'une solution convient

Mais, impossible de tester la solution complète :

- Longue durée de vie (plusieurs années)
- Matériels dédiés

—→ des **modèles** sont nécessaires.

Modélisation et analyse



Types de modèles / techniques d'analyse

Simulation :

Modèle **opérationnel**, exécutable

Test en exécutant le modèle

ex : modélisation opérationnelle des collisions : si deux paquets émis simultanément alors collision.

ex : modélisation de l'énergie en observant l'état du modèle de la radio.

Mais, pas exhaustif :

De nombreuses simulations ne permettent pas de prouver une propriété

Évaluation de performance :

Modèle **statistique**

Analyse exhaustive

ex : modélisation probabiliste des collisions : probabilité constante d'erreur.

ex : modélisation de l'énergie en comptant le nombre de messages envoyés.

Cependant, difficultés pour modéliser des comportements complexes

Propriétés intéressantes

Propriétés fonctionnelles (comme pour d'autres systèmes)

Propriétés liées à l'énergie (non-fonctionnelles) :

- “**La durée de vie du réseau est au moins de t unités de temps**”,
Définitions possibles pour la *durée de vie du réseau* :
 - instant où le premier nœud meurt
 - instant où tous les nœuds n'ont plus d'énergie
 - instant où le réseau est déconnecté
- “**Un certain nœud ne consomme pas plus de E unités d'énergie en moins de t unités de temps**”

Quelques simulateurs

Non dédiés aux réseaux de capteurs (NS-2, OpNet) :

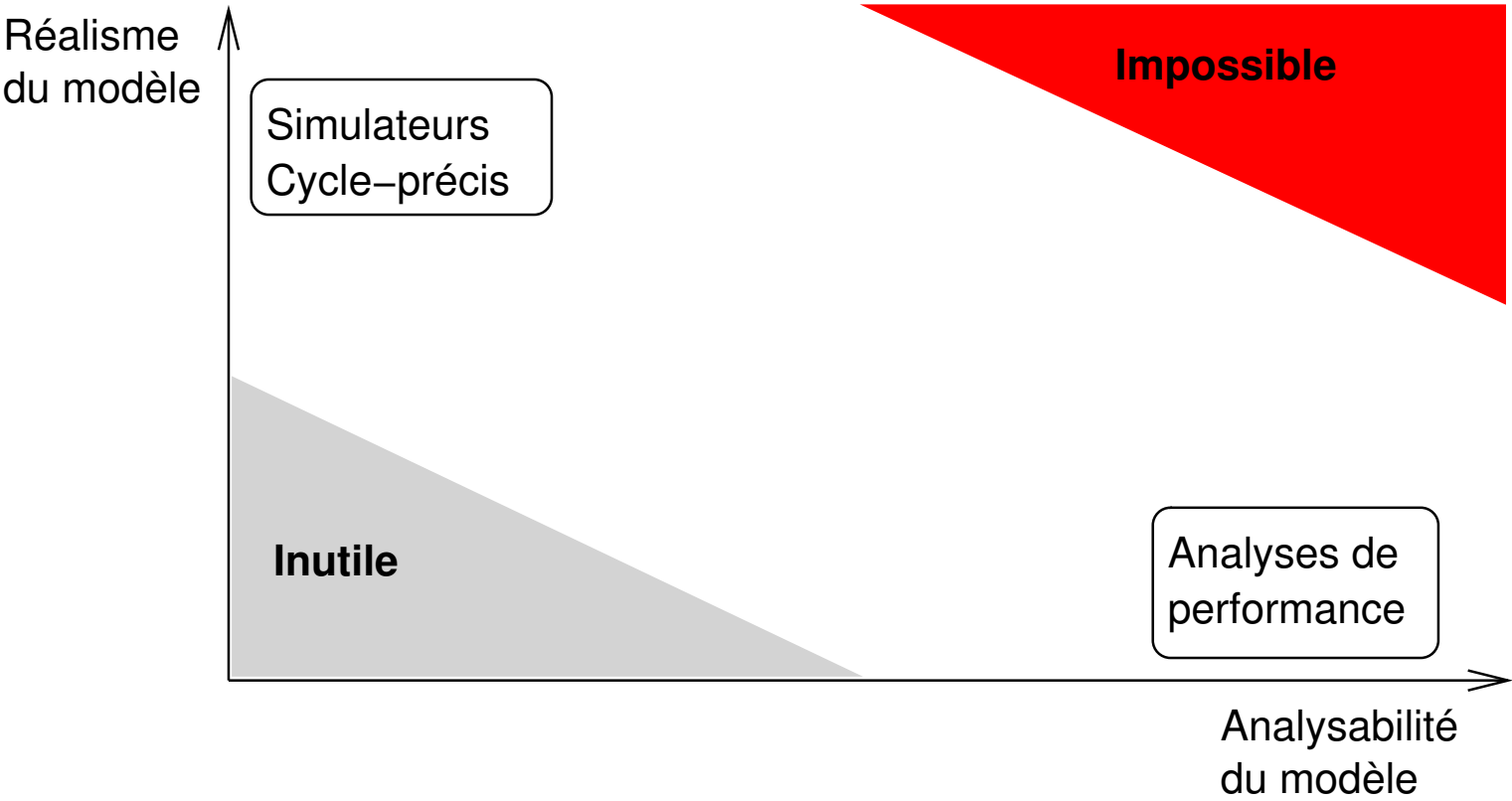
- Pas de modélisation précise de la consommation (pas de modèle du matériel)
- Pas globaux

Simulateurs **dédiés aux réseaux de capteurs** :

- Atemu, Avrora, WSNnet, PowerTOSSIM.

Modélisation (**trop ?**) précise du processeur

Panorama



Notre approche

Modèles opérationnels qui puissent être analysés formellement.

1. Estimation correcte de la consommation

→ nécessite des **modèles précis**.

Ex : simulations

2. Exploration exhaustive

Problème d'explosion d'états :

p états par nœud $\Rightarrow p^N$ états pour un réseau de N nœuds

→ nécessite des **modèles abstraits**.

3. Définition d'un cadre formel pour faire des abstractions *correctes*.

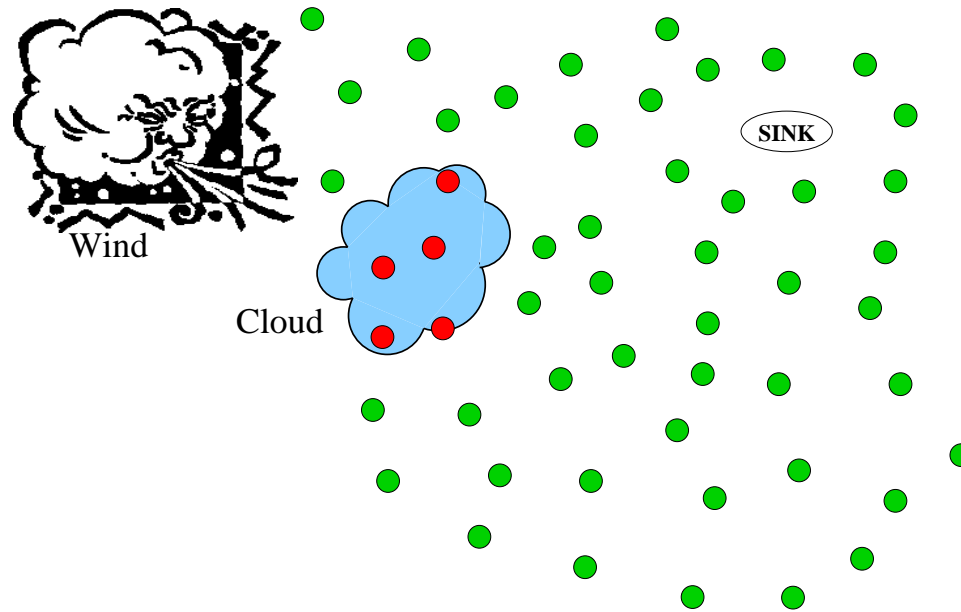
Une abstraction correcte, c'est :

propriété vraie sur le modèle abstrait \Rightarrow propriété vraie (sur le modèle détaillé)

Plan

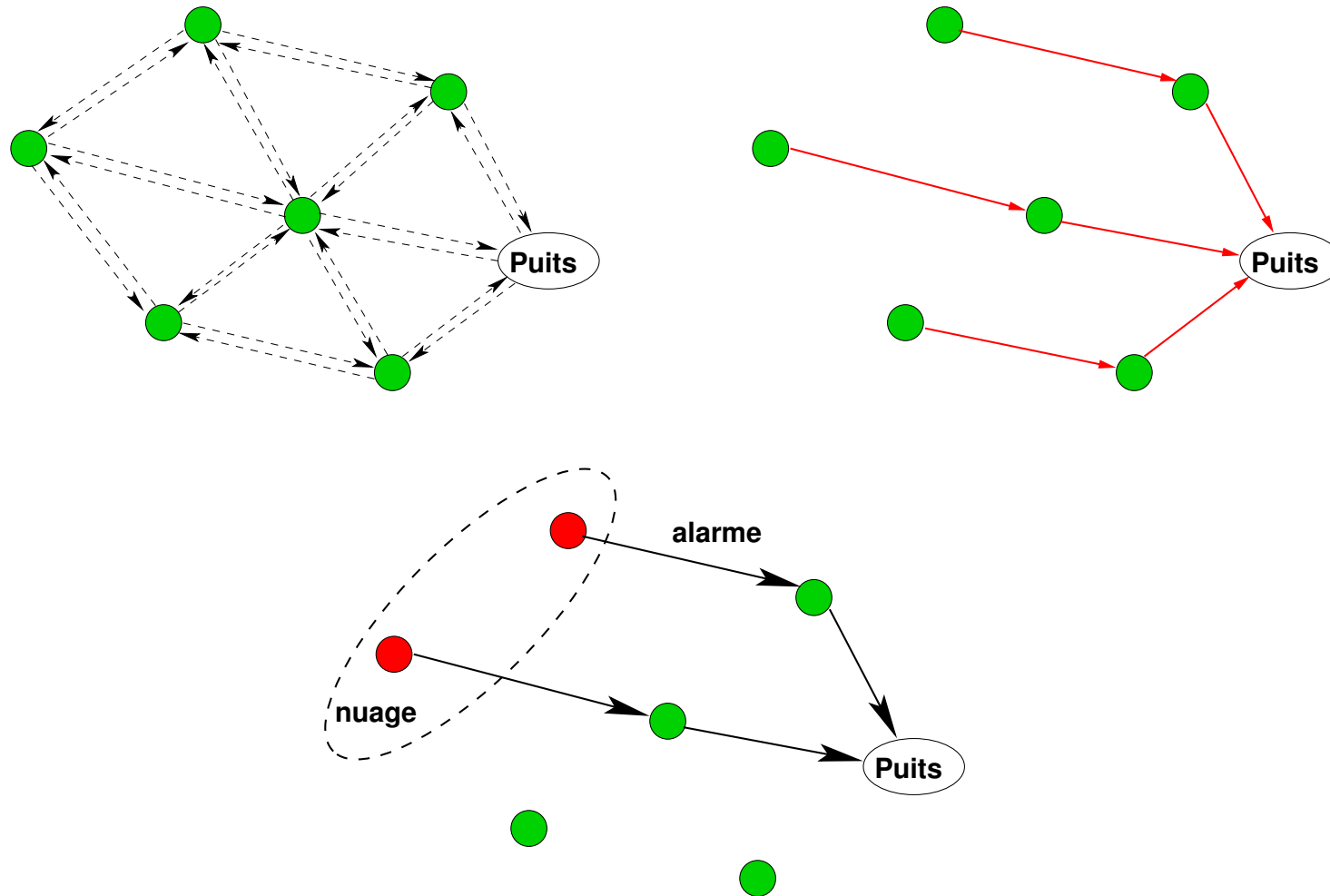
1. Problématique
2. Étude de cas
3. Modélisation détaillée : GLONEMO, un simulateur de réseaux de capteurs
4. Modélisation abstraite : analyse exhaustive d'un réseau de capteurs
5. Une notion d'abstraction pour des modèles de consommation
6. Conclusion et perspectives

Étude de cas



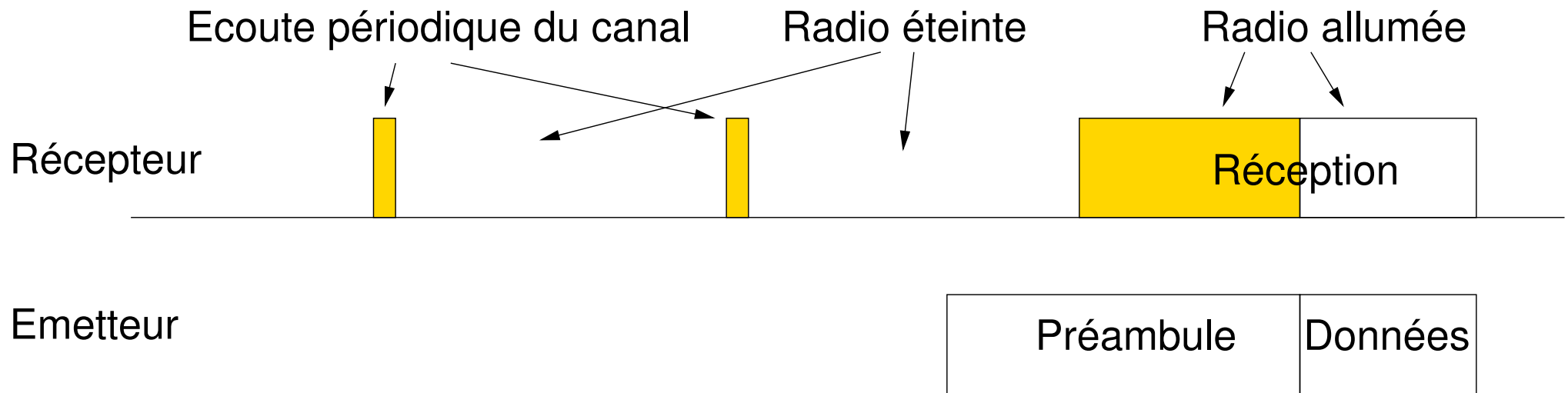
- **Application** : Détection d'un nuage toxique
- **Environnement** : Un nuage qui se déplace sous l'influence du vent.
- **Routage** : Diffusion dirigée
- **Protocole MAC** : Un protocole à échantillonnage de préambule
- **Matériel** : processeur et radio basse consommation

Le protocole de routage, diffusion dirigée



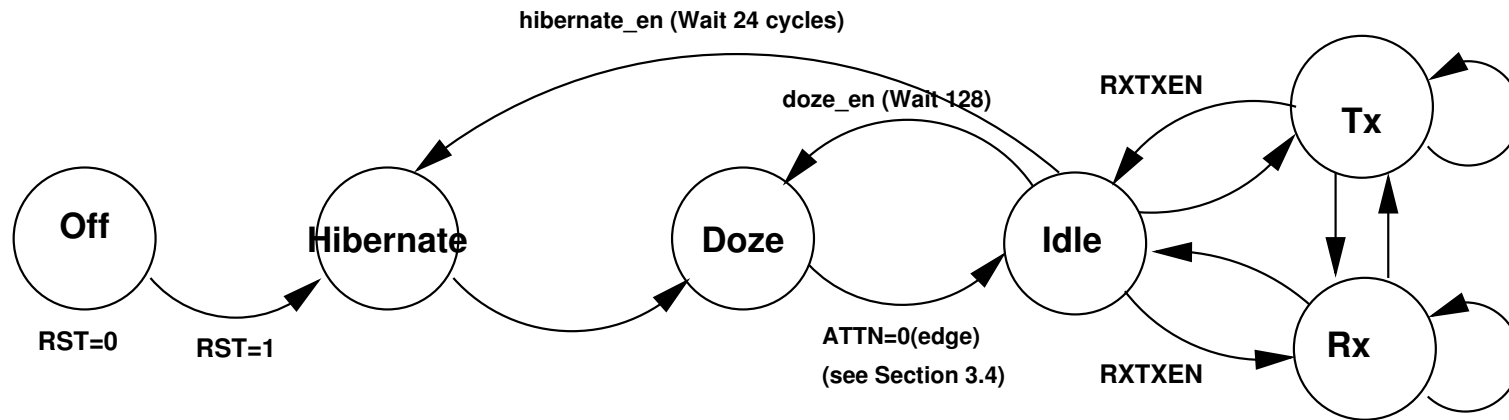
→ Besoin de modéliser du **logiciel**

Le protocole MAC (Medium Access Control), un protocole à échantillonnage de préambule :



→ Besoin de modéliser le **temps**

La radio



Plusieurs états pour le programmeur, dont :

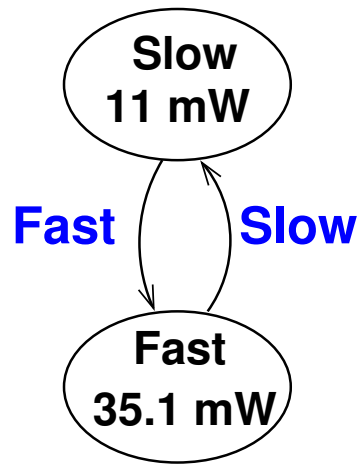
- **Off** : radio éteinte
- **Tx** : la radio émet
- **Rx** : la radio reçoit
- **Idle** : la radio allumée, prête à recevoir

→ modélisation assez fine de l'énergie nécessaire mais inutile de modéliser la physique des choses !

Microprocesseur synchrone

Peut avoir plusieurs états de consommation

Exemple :



Plan

1. Problématique
2. Étude de cas
3. Modélisation détaillée : GLONEMO, un simulateur de réseaux de capteurs
4. Modélisation abstraite : analyse exhaustive d'un réseau de capteurs
5. Une notion d'abstraction pour des modèles de consommation

Besoins pour un modèle global

- **Temps**
- **Logiciel**
- **Matériel**
- **Parallélisme :**
les nœuds s'exécutent en parallèle
- **Communication :**
entre les nœuds et au sein d'un nœud

Formalisme de GLONEMO

Modèle **réactif synchrone** :

- temps **discret** global (instants logiques)
- exécution **en parallèle** des processus
- à chaque instant, les processus réagissent

GLONEMO, un **simulateur à pas fixes** :

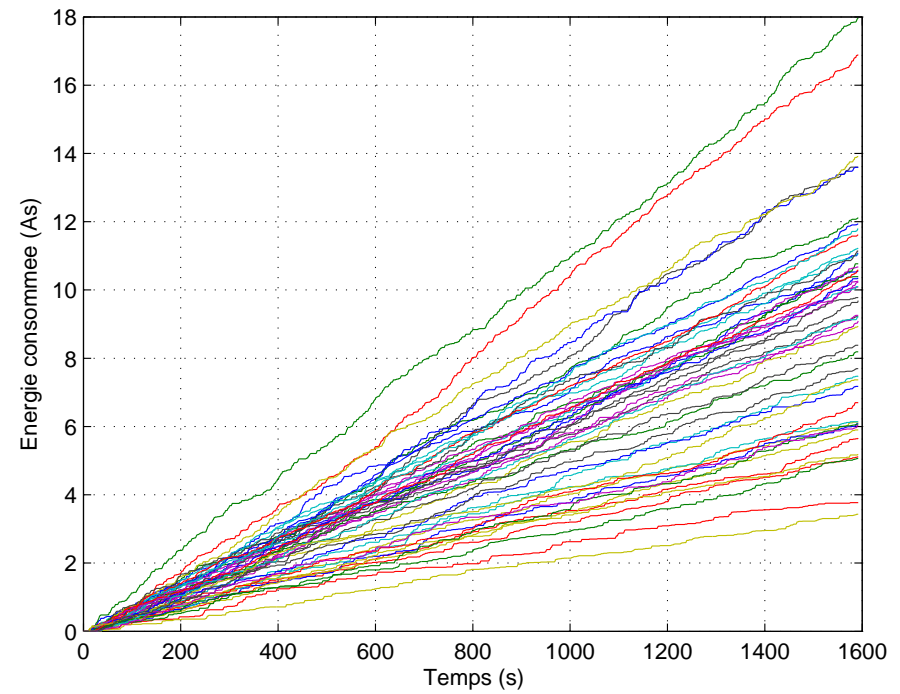
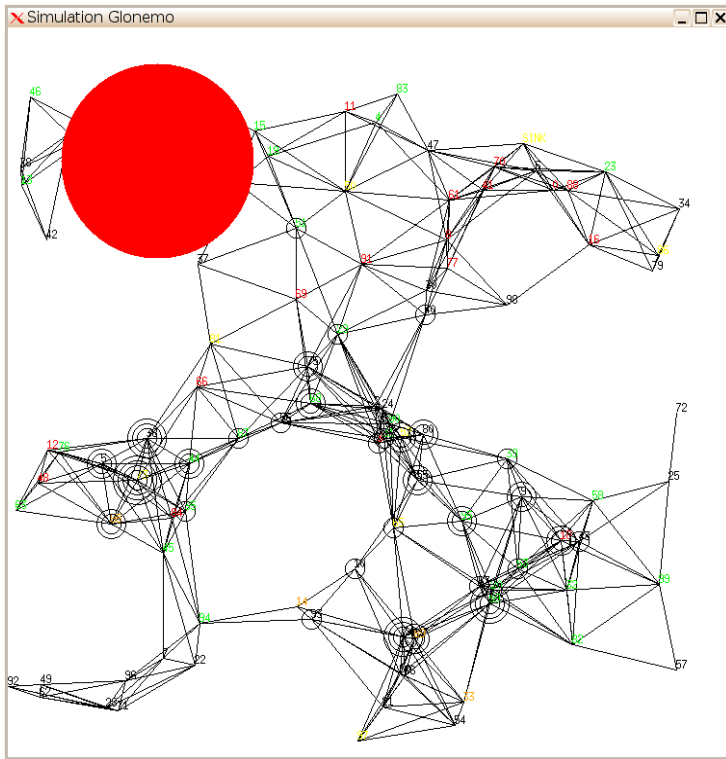
- durée entre 2 **instants logiques** = **durée fixe** de temps simulé
→ pratique pour calculer la consommation.
- **1 élément** (mac, radio ...) = **1 processus**
8 processus par nœud
- une simulation = n nœuds en parallèle

Implémenté en REACTIVEML (Louis Mandel, LRI)

- langage réactif synchrone
- langage fonctionnel à la ML

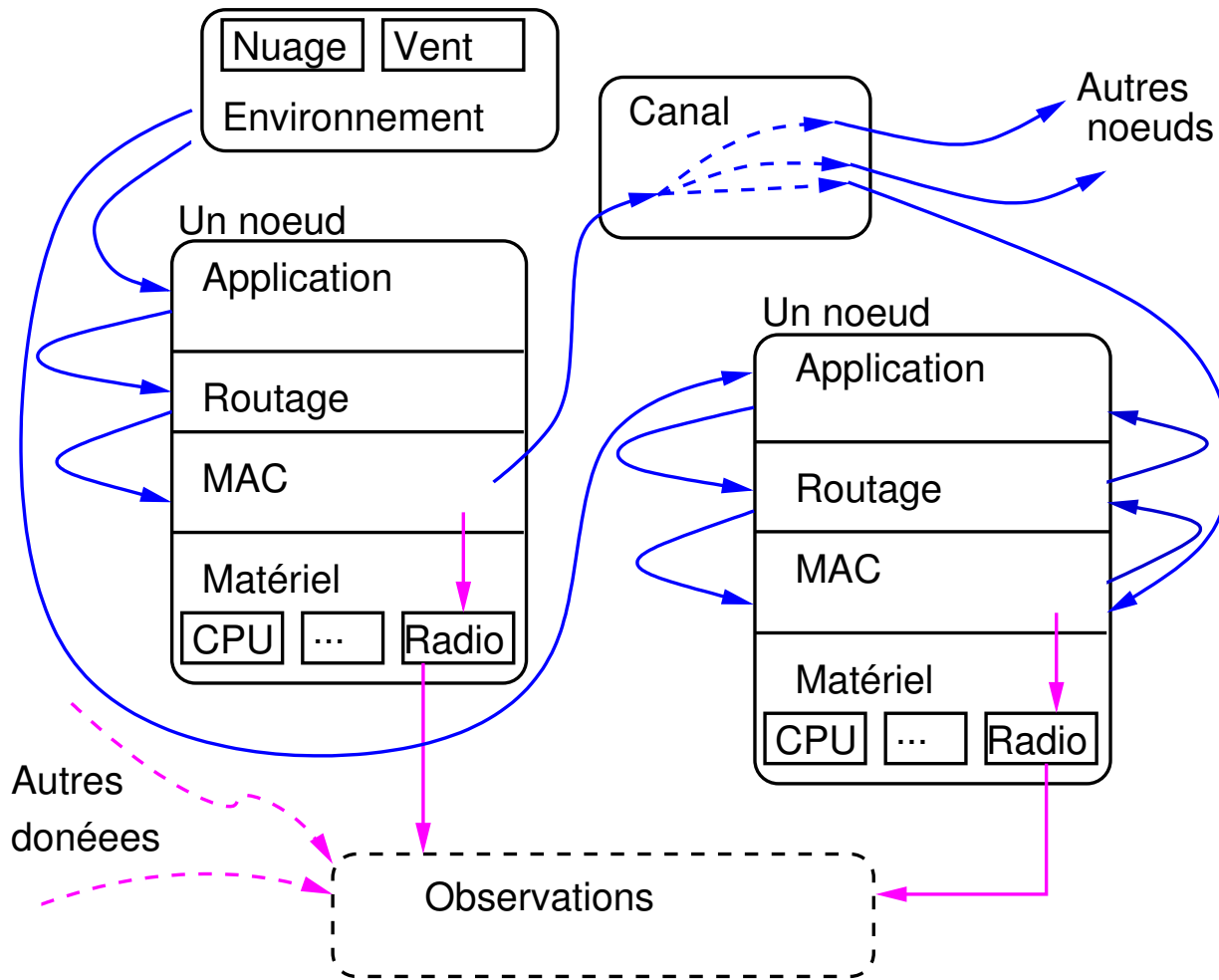
GLONEMO : GLObal NETwork MOdel

GLONEMO :



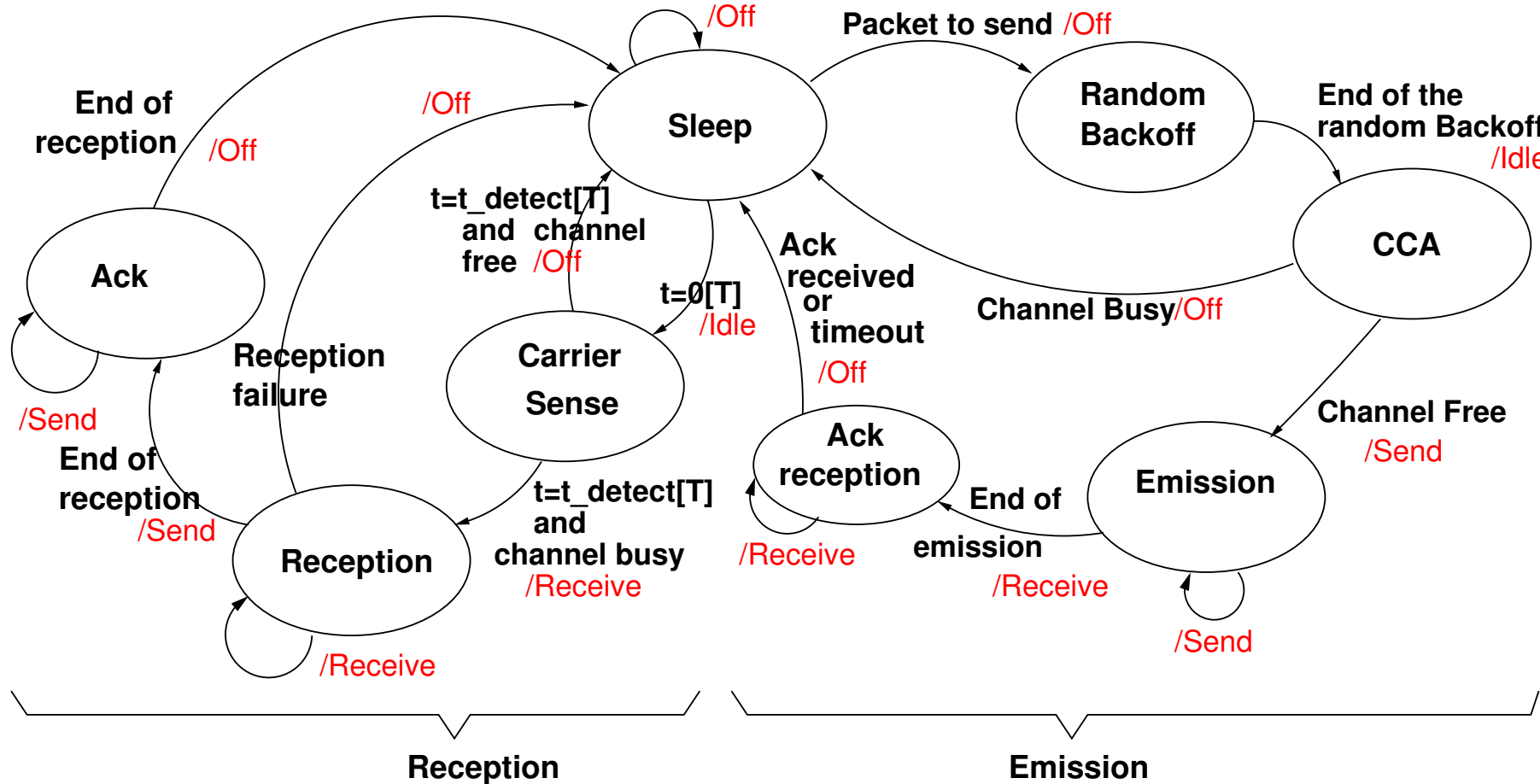
`./glonemo`

Structure du modèle



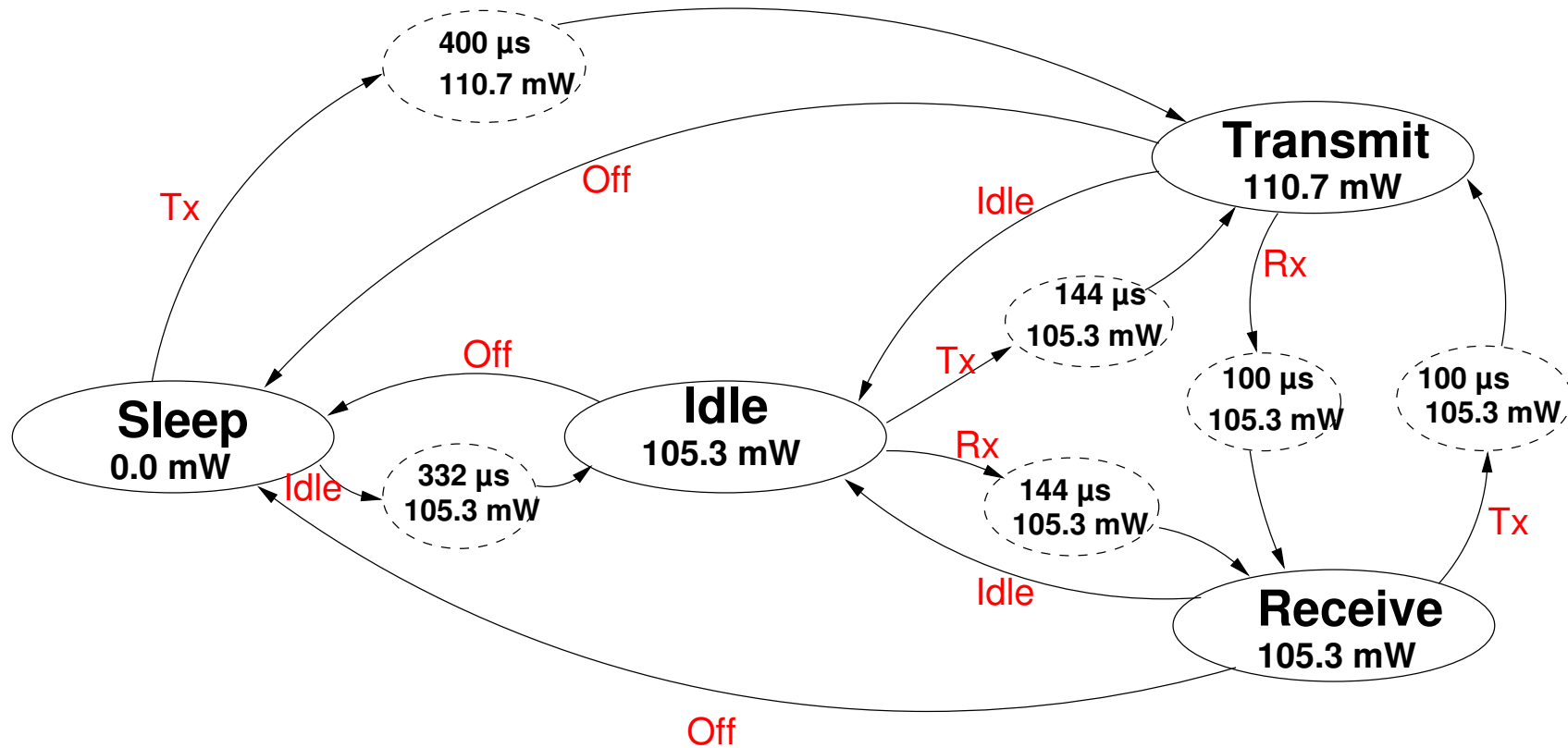
Processus synchronisés s'exécutant en parallèle

Modèle du protocole MAC



CCA : Clear Channel Assessment

Modèle de consommation de la radio



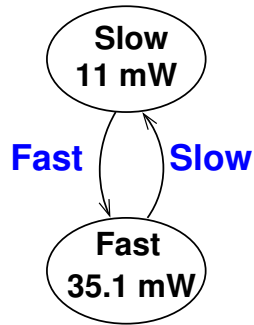
Valeurs mesurées sur le Motorola MC13192

Le modèle du **protocole MAC** pilote cet automate.

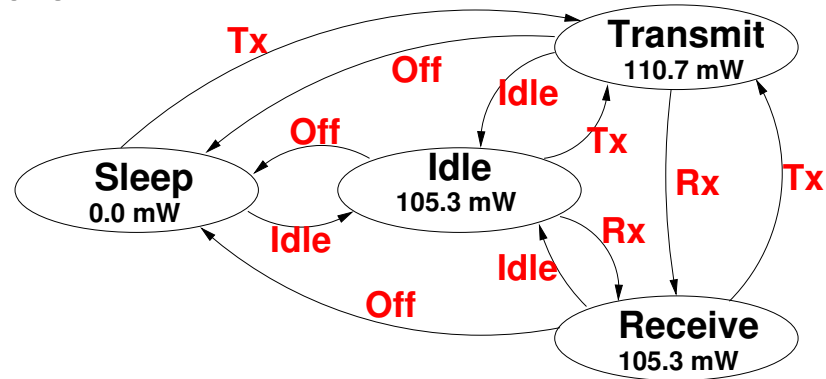
Un **"observateur"** calcule l'énergie consommée en fonction de l'état de l'automate.

Calcul de l'énergie

CPU



Radio



Radio :	entrées états conso (mW)	Idle Sleep 0.0	- Idle 105.3	Tx Idle 105.3	- Tx 110.7	- Tx 110.7	Off Tx 110.7	- Sleep 0.0
CPU :	entrées états conso (mW)	Fast Slow 11	- Fast 35.1	- Fast 35.1	- Fast 35.1	- Fast 35.1	Slow Fast 35.1	- Slow 11
Composition	(mW)	11	140.4	140.4	145.8	145.8	145.8	11
1 inst. = 10^{-2} s	mJ	0.11	1.404	1.404	1.458	1.458	1.458	0.11
intégration	mJ	0.11	1.514	2.918	4.376	5.834	7.292	7.402

Modèle de l'environnement

2 processus `wind` et `cloud` :

– communiquent : `wind` → `cloud`

– **non déterministes** et **contraints**

(ex : `cloud` est contraint par `wind` mais plusieurs évolutions possibles).

Implémentés grâce à LUCKY (E. Jahier, P. Raymond, VERIMAG) :

– permet de décrire et d'exécuter des systèmes réactifs non-déterministes

– connecté à REACTIVEML

Importance du modèle d'environnement, comparaison de simulations :

- avec un modèle d'environnement basé sur des lois de Poisson
- avec notre modèle plus réaliste : stimuli **corrélés temporellement et spatialement**
→ Résultats complètement différents !

Des stimuli décorrélés sont optimistes : moins de collisions

GLONEMO, conclusion

Bilan :

- **Modélisation opérationnelle de la consommation**
- Un simulateur **efficace** : simule plusieurs dizaines de milliers de nœuds plus rapide que le temps réel jusqu'à 1000 nœuds
- Simulations **réalistes** grâce au **modèle d'environnement**

De plus,

- **REACTIVEML** convient pour programmer des simulateurs dédiés
- Base pour construire des modèles de réseaux de capteurs dans d'autres formalismes

Mais, à cause de la taille du modèle, seule la simulation est efficace

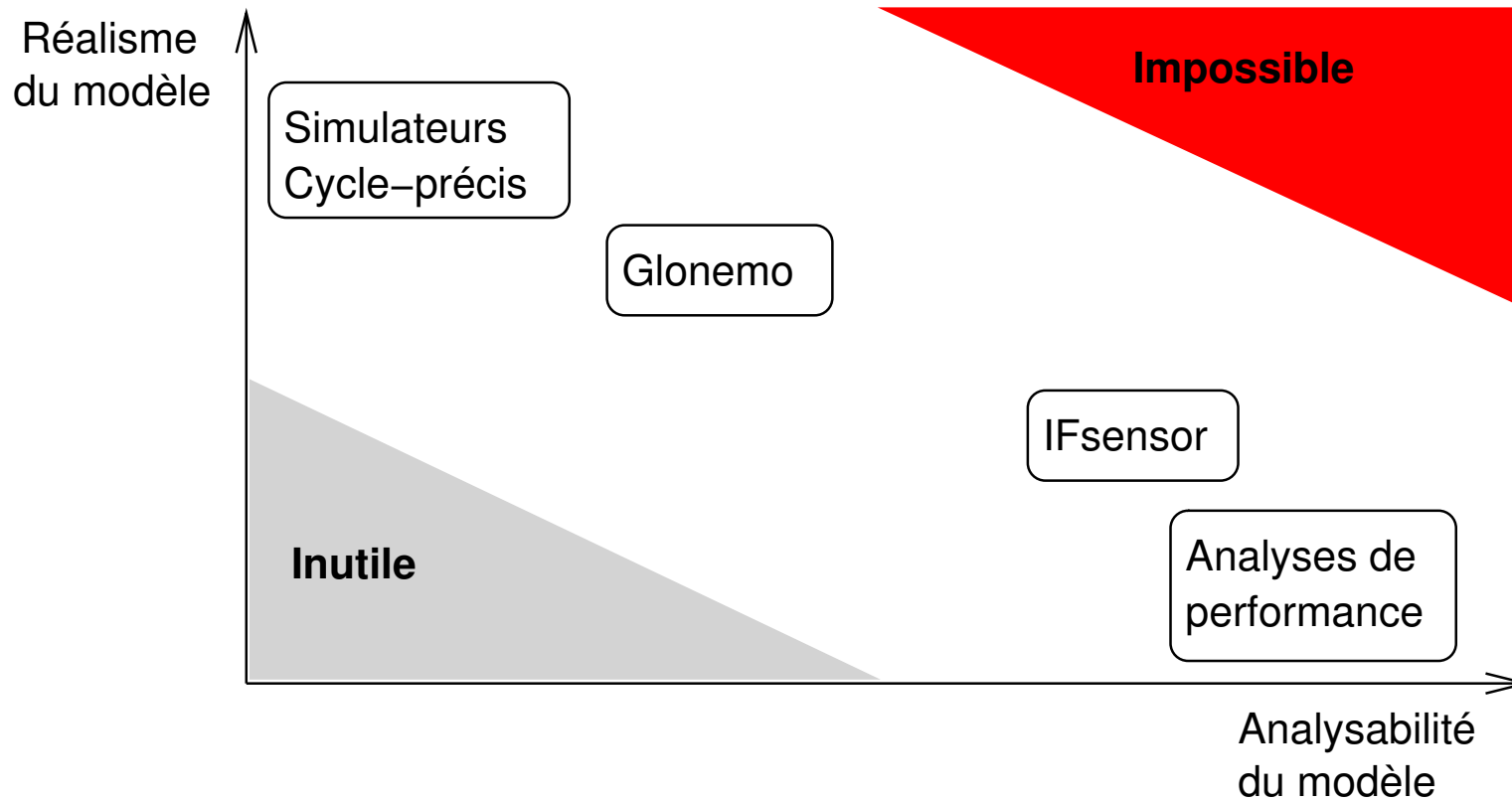
À suivre,

Nous proposons maintenant un **modèle abstrait**

Plan

1. Problématique
2. Étude de cas
3. Modélisation détaillée : GLONEMO, un simulateur de réseaux de capteurs
4. Modélisation abstraite : analyse exhaustive d'un réseau de capteurs
5. Une notion d'abstraction pour des modèles de consommation
6. Conclusion et perspectives

Notre approche

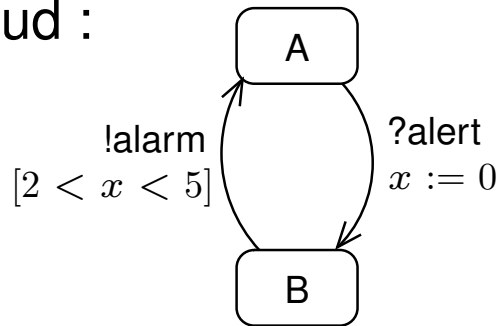


Le formalisme IF

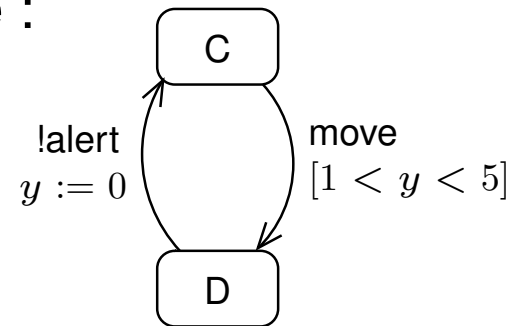
- Automates temporisés
- Parallélisme *asynchrone*
- *Communication* par échange de messages

Exemple :

Nœud :



Nuage :



Exemple de trace pour le produit :

$$\begin{aligned}
 & (A, C, x = 0, y = 0) \xrightarrow{\delta(2)} (A, C, x = 2, y = 2) \xrightarrow{\text{move}} (A, D, x = 2, y = 2) \xrightarrow{\text{!alert}} (A, C, x = 2, \\
 & y = 0) \xrightarrow{\text{?alert}} (B, C, x = 0, y = 0) \xrightarrow{\delta(4)} (B, C, x = 4, y = 4) \xrightarrow{\text{!alarm}} (A, C, x = 4, y = 4)
 \end{aligned}$$

IF permet de calculer le **graphe d'état** qui représente **tous les comportements**.

IF : équipe DCS, Verimag

Notre étude de cas

Nous étudions des protocoles de routage :

Comparaison des algorithmes de diffusion dirigée et inondation.

Les autres éléments sont plus abstraits :

- Application : détection d'un nuage toxique
- MAC : des délais et collisions peuvent se produire
- Environnement : un nuage

Notre modélisation : IFSENSOR

Durée de vie : une horloge globale, *lifetime*

protocole MAC et canal radio :

→ Réalité :

- Unicast : acquittement
- Broadcast : pas d'acquittement possible

→ Notre modèle :

- Unicast : durée d'émission variable
- Broadcast : pertes non-déterministes. durée d'émission fixe.

Énergie : uniquement sur les transmissions.

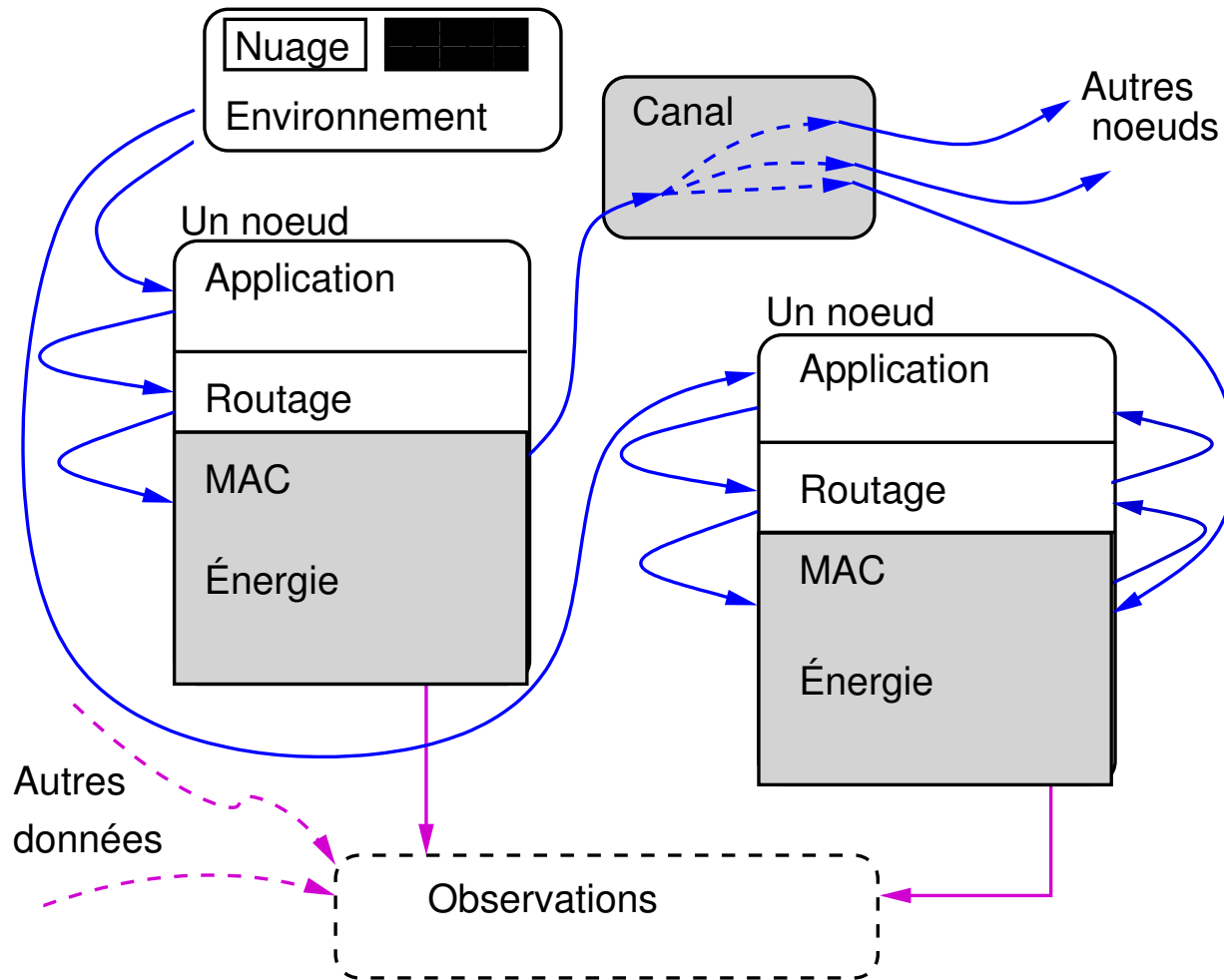
une émission = 3 unités d'énergie

une réception = 2 unités d'énergie

→ **Toute la mécanique de transmission est abstraite**

Environnement : périodiquement, le nuage stimule un nœud choisi aléatoirement parmi les voisins du nœud précédent.

Structure du modèle



Calcul de la durée de vie du scénario pire-cas

Scénario pire-cas = durée de vie la plus courte

Plusieurs critères pour la durée de vie, par exemple :

- premier nœud mort
- réseau déconnecté

→ permet d'identifier les états du système où le réseau ne vit plus (états **DEAD**).

Une exécution possible = un chemin du graphe d'états

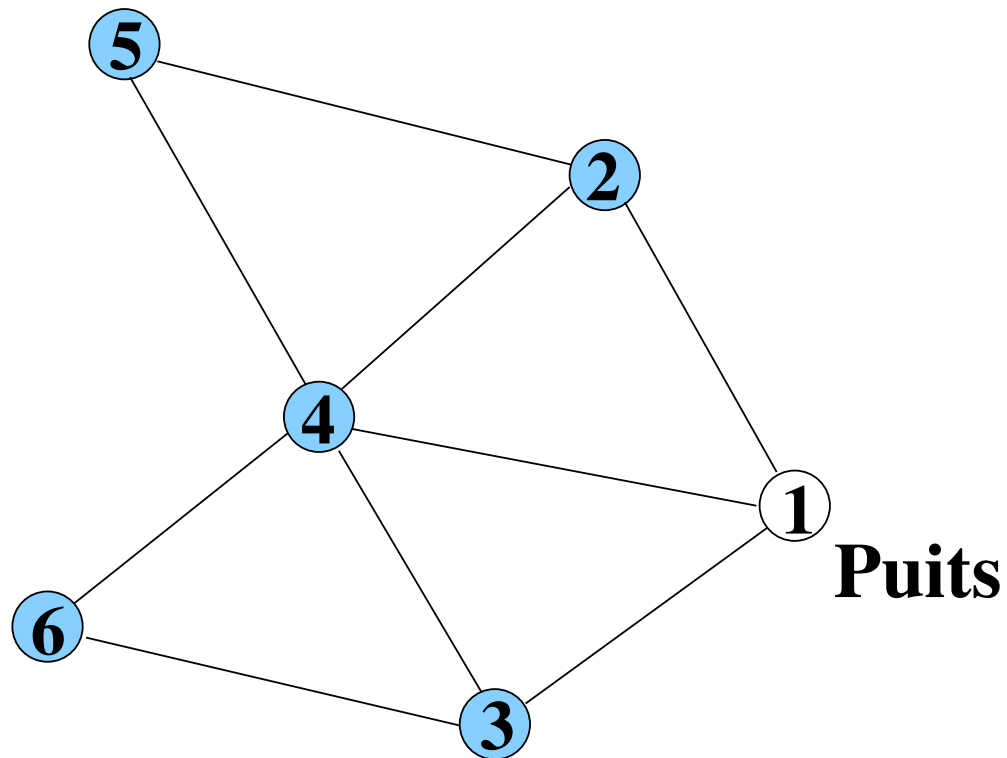
Le scénario pire-cas = le plus court chemin (selon l'horloge *Lifetime*) d'un état initial à un état **DEAD**.

Mise en œuvre :

- les outils IF pour explorer le graphe d'état
- et, un algorithme de recherche de plus court chemin (A^*) (modifié).

Quelques résultats

Nous avons considéré ce réseau de six nœuds :



Quelques résultats

1. Résultats conformes aux prévisions : diffusion dirigée mieux que inondation

	1 ^{er} nœud mort	Réseau déconnecté
Inondation	14	21
Diffusion dirigée	41	52

Durées de vie pire-cas

2. Résultats différents de simulations :

	Pire-cas	Exemples de résultats obtenus par simulation				
Inondation	14	40	32	18	21	19
Diffusion dirigée	41	80	78	123	108	101

Durée de vie = 1^{er} nœud mort

- Énergie : émission = 3, réception = 2 et capacités des batteries = 40 unités d'énergie
- Une émission dure de 1 à 3 unités de temps et 5 de plus si retransmission
- Le nuage apparaît toutes les 7 unités de temps

Taille du modèle

nb de nœuds	4	6	8	10
nb d'états explorés	250 000	1 900 000	2 300 000	2 800 000
nb d'horloges ($2 + N$)	6	8	10	12
temps de calcul	2 min	38 min	10 h	2 jours

Il faudrait encore simplifier le modèle

IFSENSOR, conclusion

Bilan :

- IF convient pour une **modélisation abstraite d'un réseau de capteurs**
- Les outils permettant de trouver le **scénario pire-cas** existent

Mais,

- modèle pas assez abstrait
- et abstractions non justifiées

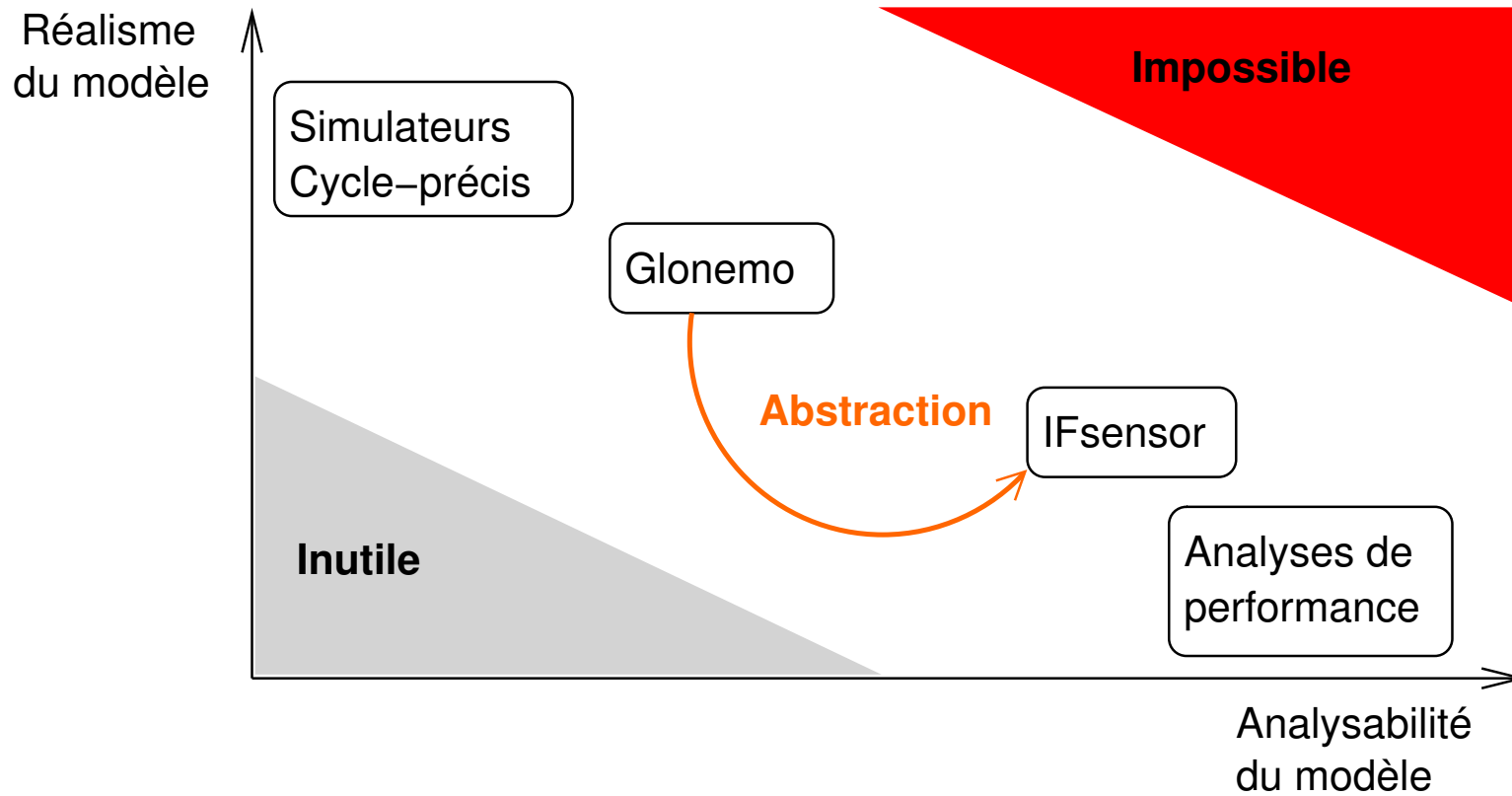
À suivre :

- Un cadre formel pour construire des **abstractions correctes** est nécessaire !
Une abstraction correcte, c'est :
propriété vraie sur le modèle abstrait \Rightarrow propriété vraie (sur le modèle détaillé)

Plan

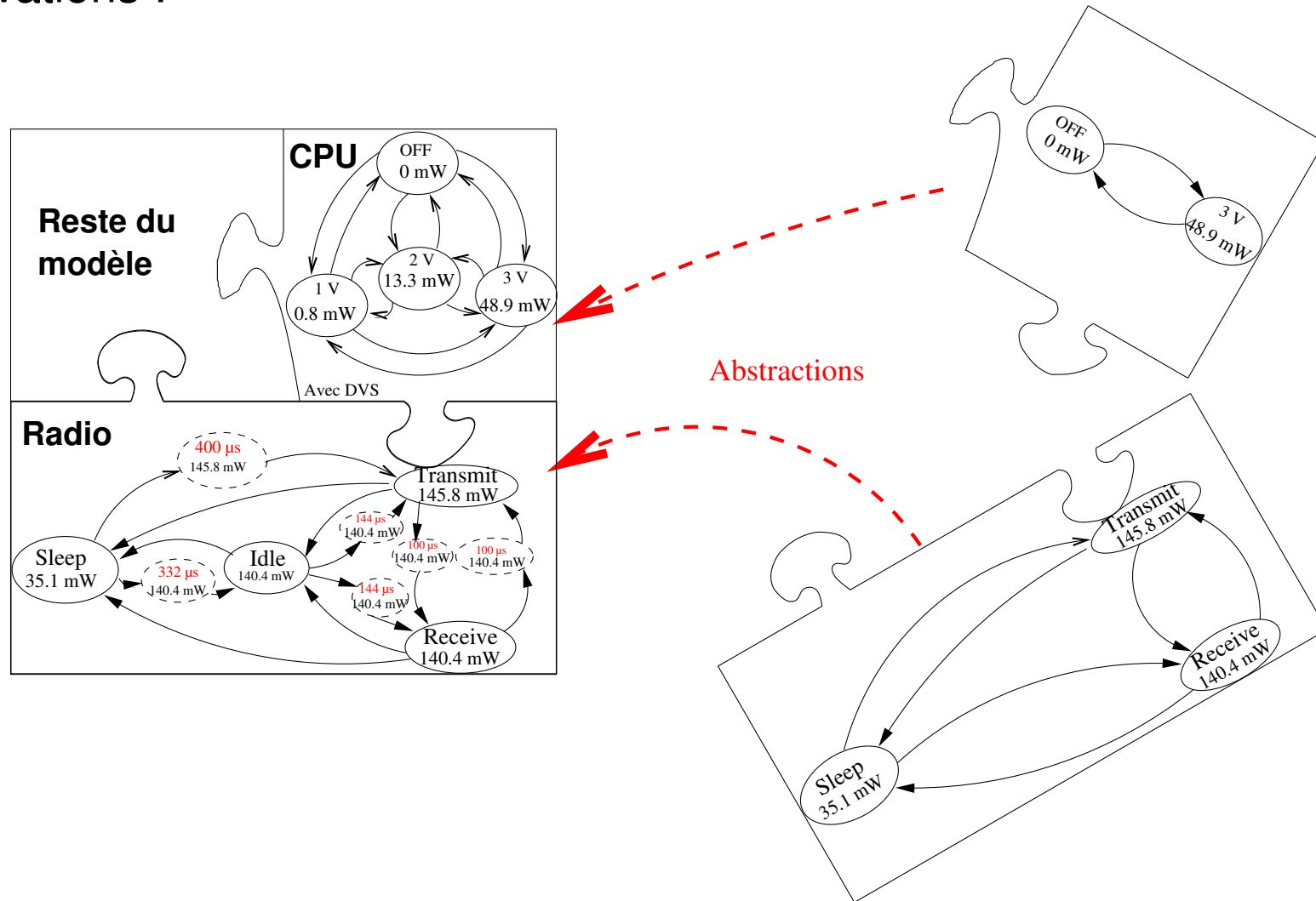
1. Problématique
2. Étude de cas
3. Modélisation détaillée : GLONEMO, un simulateur de réseaux de capteurs
4. Modélisation abstraite : analyse exhaustive d'un réseau de capteurs
5. Une notion d'abstraction pour des modèles de consommation
6. Conclusion et perspectives

Notre approche



Abstraction pour des modèles de consommation

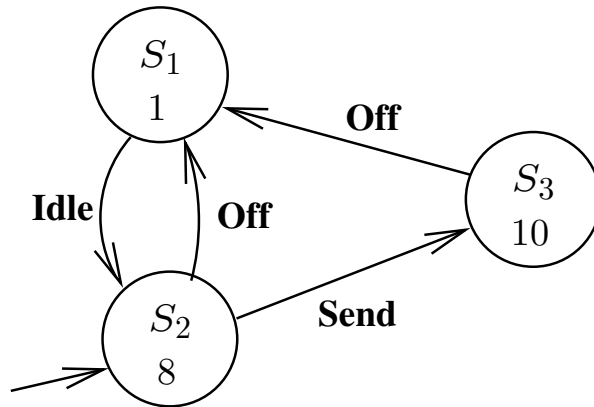
Motivations :



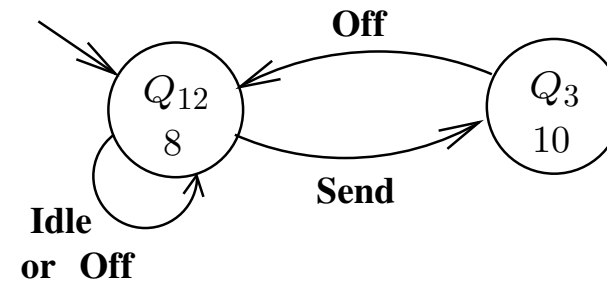
Exemple : abstraction brutale

Fonctionnellement, les deux modèles doivent être équivalents (mêmes sorties).

Non-fonctionnellement, le modèle abstrait consomme plus que le modèle détaillé on s'intéresse à des propriétés pire-cas.



(a) Modèle détaillé, R_1 .

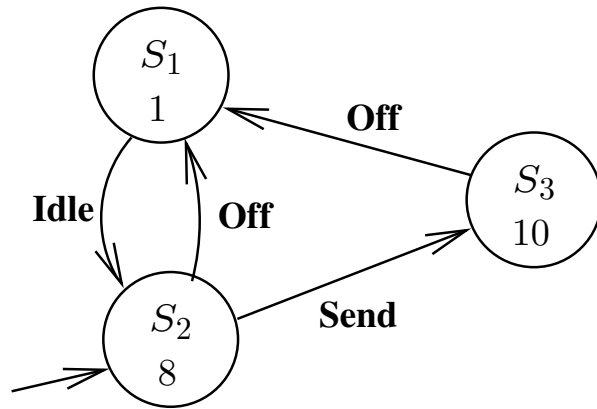


(b) Modèle abstrait, R_2 .

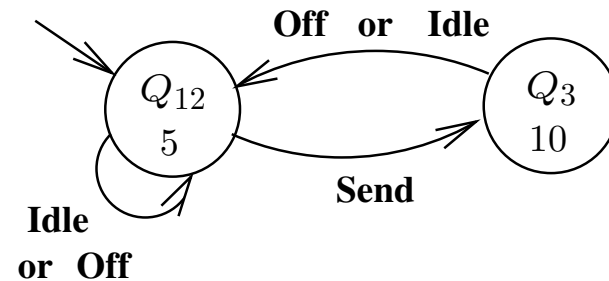
$$\forall e \text{ entrées, } \text{conso}(R_1, e) \leq \text{conso}(R_2, e)$$

→ **Mais**, abstraction très (trop ?) brutale

Exemple d'abstraction



(c) Modèle détaillé, R_1 .



(d) Modèle abstrait, R_2 .

Abstraction plus fine, **mais** n'est **pas toujours vraie** :

$\exists e$ telle que $\text{conso}(R_1, e) \not\subseteq \text{conso}(R_2, e)$

Idée : utilisation d'un contexte K dans lequel la relation est vraie

Définition de notre abstraction, avec contexte

Utilisation d'un contexte K pour lequel la relation doit être vraie.

K est un ensemble de traces sur les entrées "Send", "Idle" et "Off".

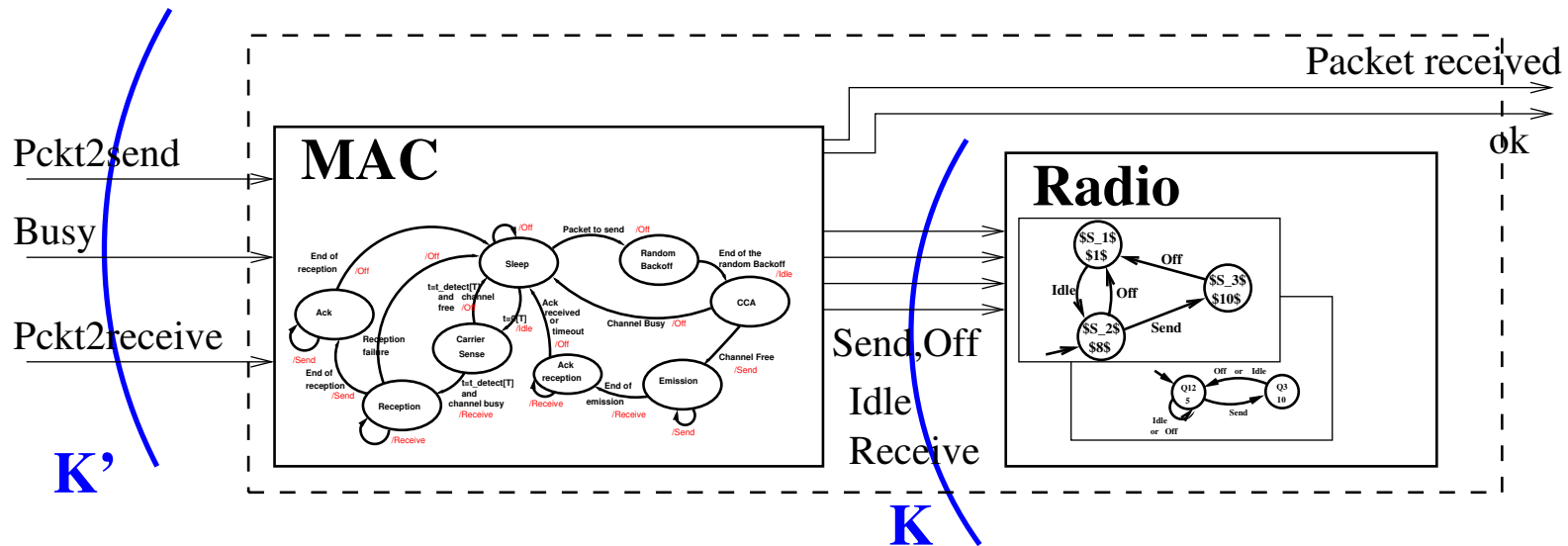
Dans le contexte K , R_2 est une abstraction de R_1 .

Définition :

$$R_1 \preceq_K R_2 \iff \forall e \in K, \begin{cases} \text{Out}(R_1, e) = \text{Out}(R_2, e) \\ \text{et} \\ \text{Conso}(R_1, e) \leq \text{Conso}(R_2, e) \end{cases}$$

→ Ces modèles se composent avec le protocole MAC.

Produit des composants



Produit synchrone : connexion flot-de-données.

Propriété (pré-congruence) :

Si, pour toutes les traces de K' , les sorties de M satisfont K **alors**

$$R_1 \preceq_K R_2 \implies R_1 \parallel M \preceq_{K'} R_2 \parallel M$$

Propriété prouvée dans le cas général.

Abstraction pour des modèles de consommation, conclusion

Bilan :

- Définition d'une relation d'abstraction
 - pour des **modèles de consommation**,
 - dans le cas **synchrone**,
- Application à un **exemple issu des réseaux de capteurs** (Radio || MAC)

De plus,

- On a prouvé la **pré-congruence** avec le **produit**

Plan

1. Problématique
2. Étude de cas
3. Modélisation détaillée : GLONEMO, un simulateur de réseaux de capteurs
4. Modélisation abstraite : analyse exhaustive d'un réseau de capteurs
5. Une notion d'abstraction pour des modèles de consommation
6. Conclusion et perspectives

Conclusions

Bilan :

- **GLONEMO**, une modélisation précise [Intersense'06, IWWAN'06]
- **IFSENSOR**, un modèle abstrait pour la vérification exhaustive [PE-WASUN'07]
- Définition d'une **relation d'abstraction** de modèles de consommation *synchrones*

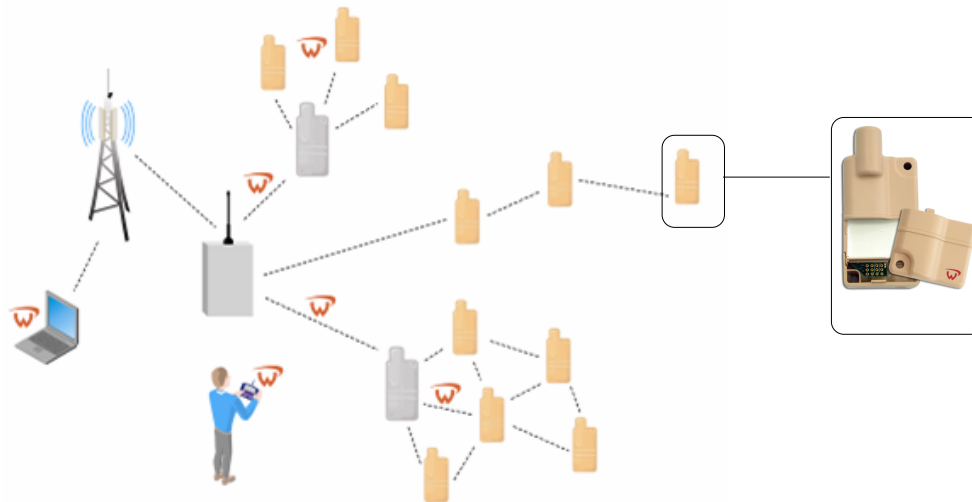
Autres travaux (non exposés) :

- **LUSSENSOR** : transformation de GLONEMO en LUSTRE [SLA++P'07]
→ pour tirer parti de la boîte à outils LUSTRE
- **Analyses de performance** de protocoles MAC [IWWAN'06, brevet]

Perspectives

GLONEMO, le simulateur global utilisé dans ARESA (O. Bezet) :

- Amélioration du modèle du canal
- Simulations globales des solutions de routage/auto-organisation
protocole de routage avec coordonnées virtuelles de T. Watteyne
- Validation de GLONEMO : modélisation d'un système déployé
Application Coronis Systems : télé-relève de compteurs



Perspectives

Modélisation formelle :

- **Automatisation** des abstractions, besoins d'algorithmes pour prouver :
 - $R_1 \preceq_K R_2$
 - « pour toutes les traces de K' , les sorties de M satisfont K »
- Application sur le modèle LUSSENSOR

Modélisation globale :

- Un formalisme globalement asynchrone et localement synchrone (**GALS**)

Modélisation abstraite souhaitée :

- Quel niveau de détail est nécessaire pour les réseaux de capteurs ?
- Comment simplifier une partie du réseau ?

Questions ?

