

Wireless Sensor Network (WSN) : Localization

December 2008

Abdallah MAKHOUL

Laboratoire d' Informatique de l'UPPA (LIUPPA)



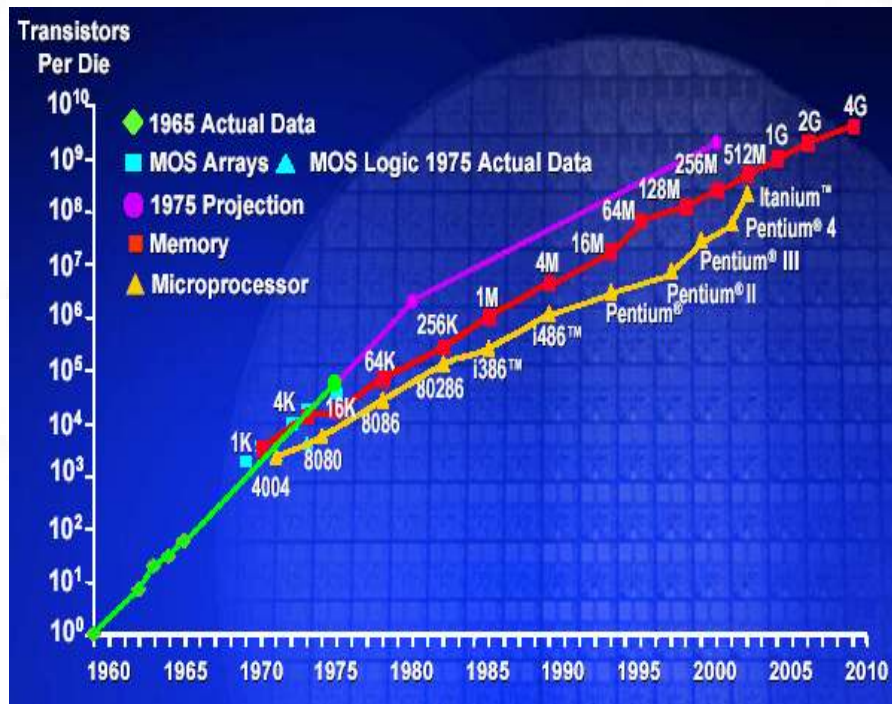
Faster, Smaller, Numerous

- **Moore's Law**

- “Stuff” (transistors, etc) doubling every 1-2 years

- **Bell's Law**

- New computing class every 10 years



Outline



**I. Wireless sensor
Networks
(Short Survey)**

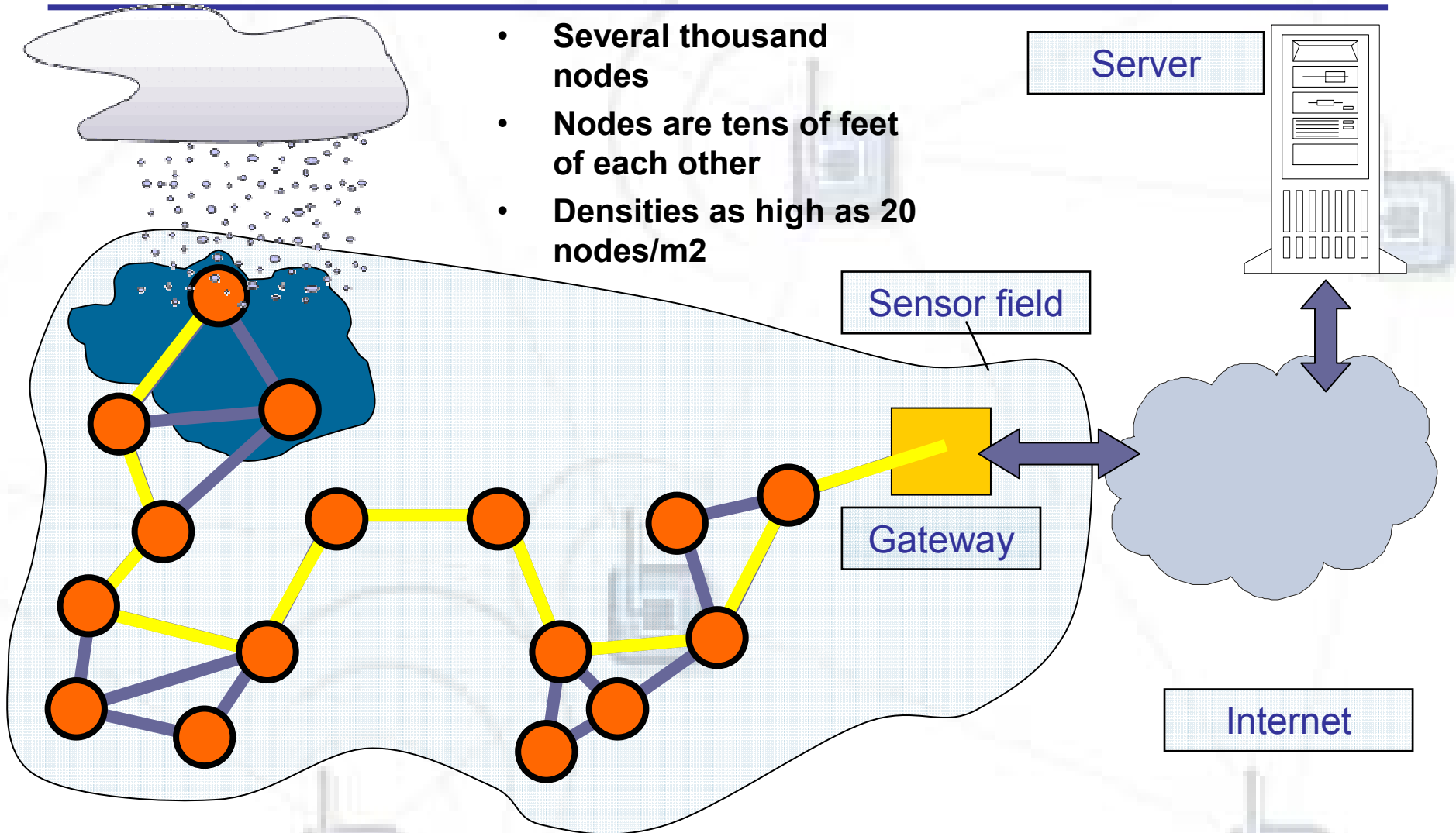
**II. Localization in
WSN**

I. Wireless Sensor Networks (Short survey)

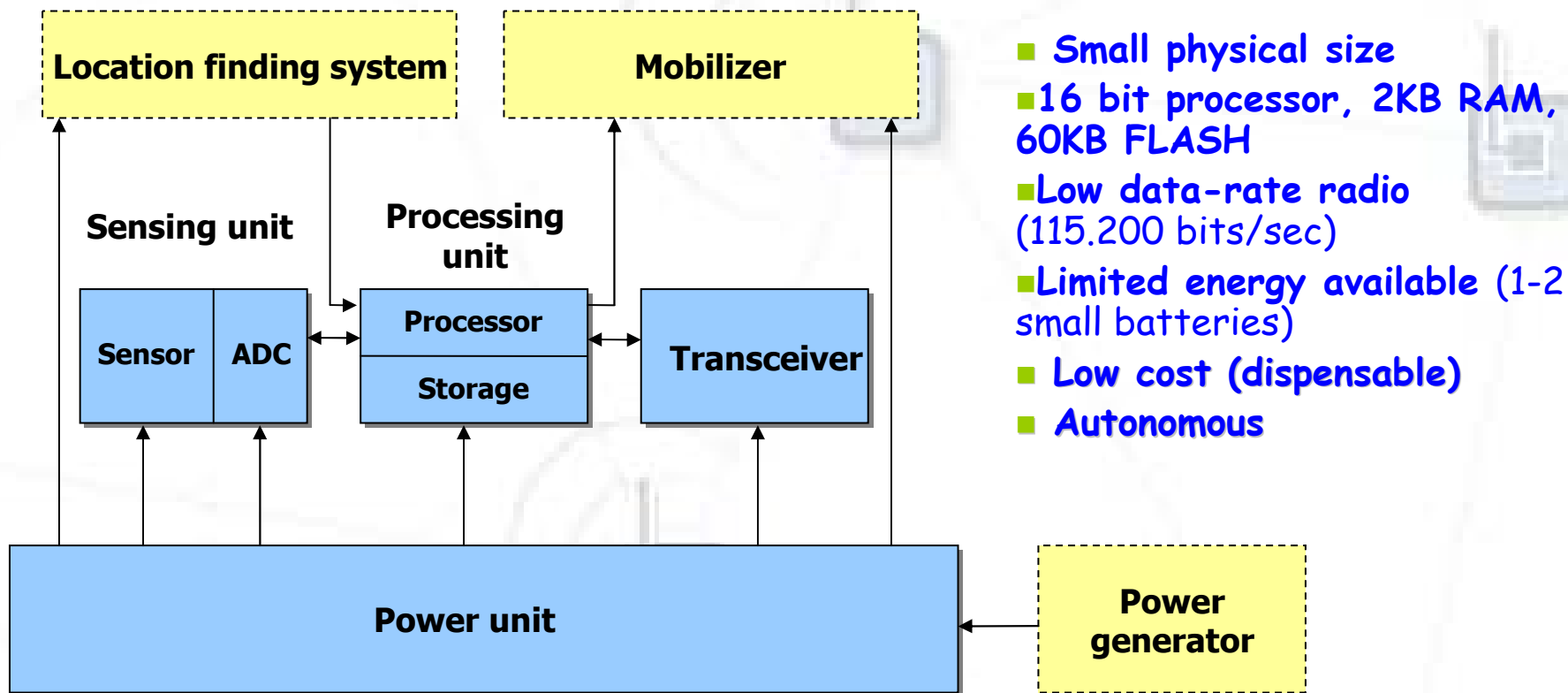
- Introduction:
 - Wireless Sensor Network (WSN)
 - Sensor node
- Applications
- WSN unique Characteristics
- Some challenges of WSN
- Conclusion I

Wireless Sensor Network

- **Several thousand nodes**
- **Nodes are tens of feet of each other**
- **Densities as high as 20 nodes/m²**



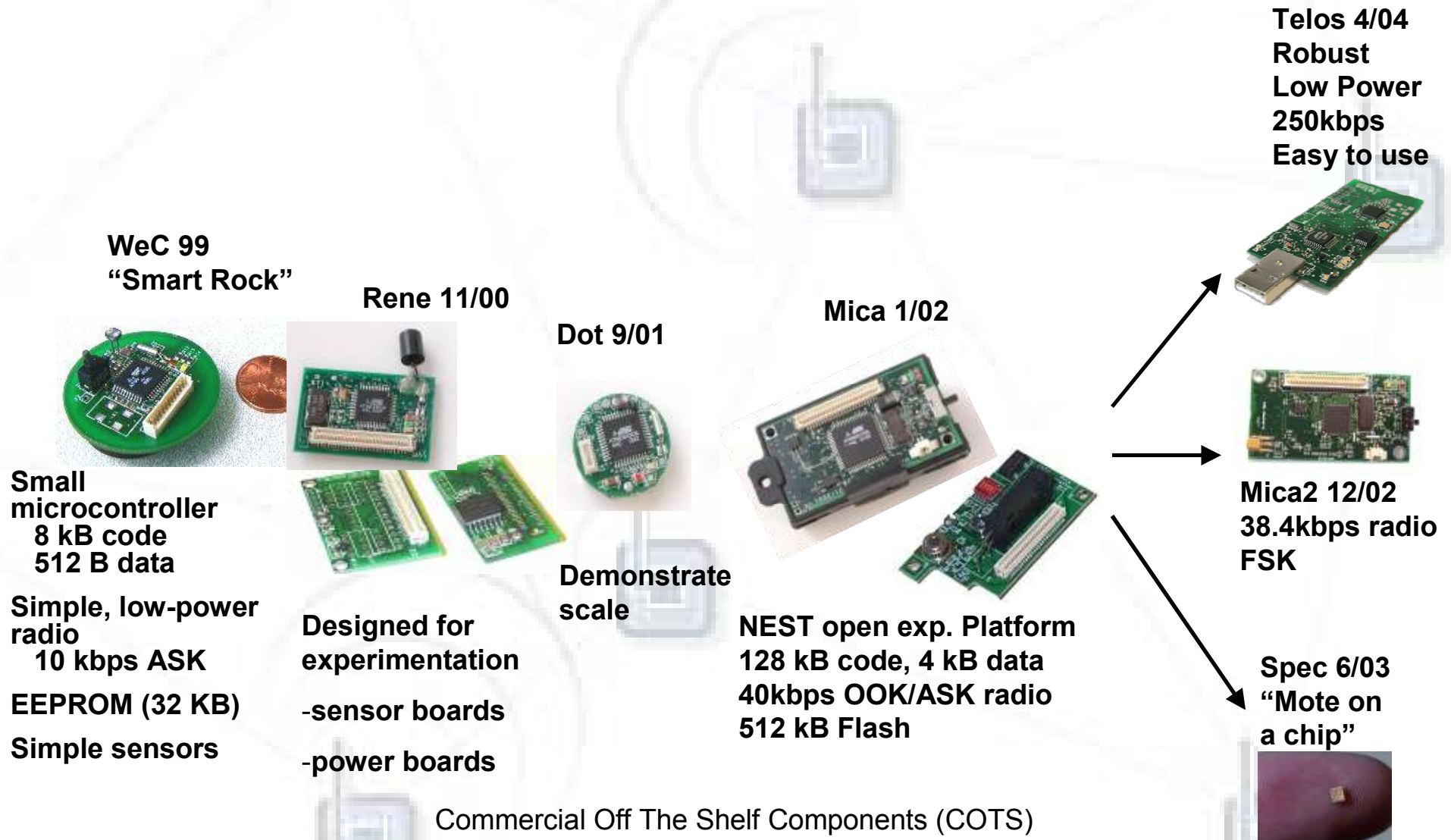
What is a mote (sensor node)?



- Small physical size
- 16 bit processor, 2KB RAM, 60KB FLASH
- Low data-rate radio (115.200 bits/sec)
- Limited energy available (1-2 small batteries)
- Low cost (dispensable)
- Autonomous

<The component of sensor node>

Sensor Node



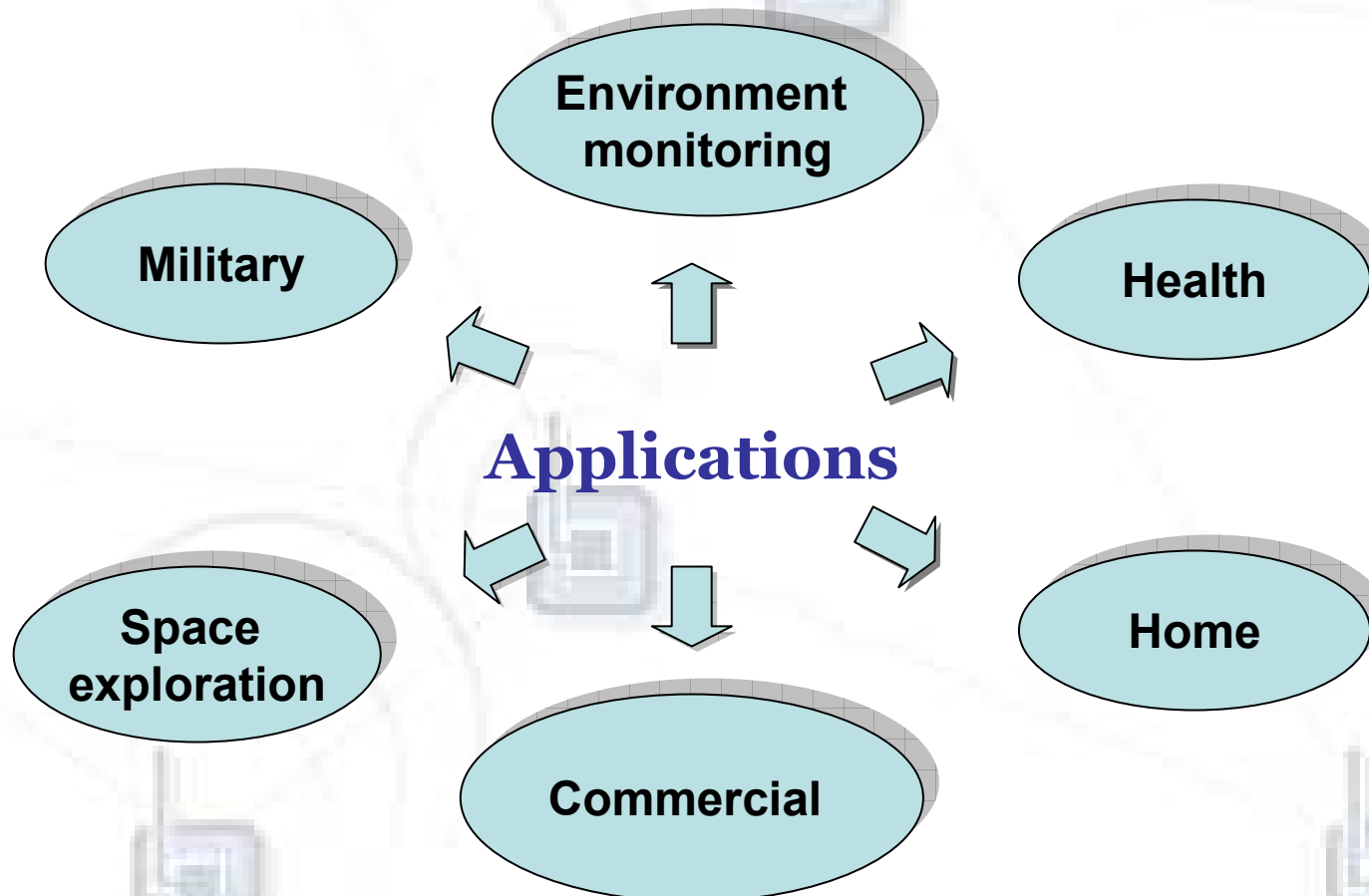


The diagram illustrates a network of six wireless sensor nodes. Each node is represented by a small grey square with a vertical antenna on top. The nodes are interconnected by thin grey lines, forming a network topology. Concentric circles around each node represent their communication ranges. The text 'WSN Applications' is centered in the image in a bold, blue, sans-serif font.

WSN Applications

Sensor Network Applications

Sensing + CPU + Radio => Thousands of potential applications



Sensor Network Applications (examples)



Volcano Monitoring

<http://www.eecs.harvard.edu/~werner/projects/volcano/>



Reindeer Tracking

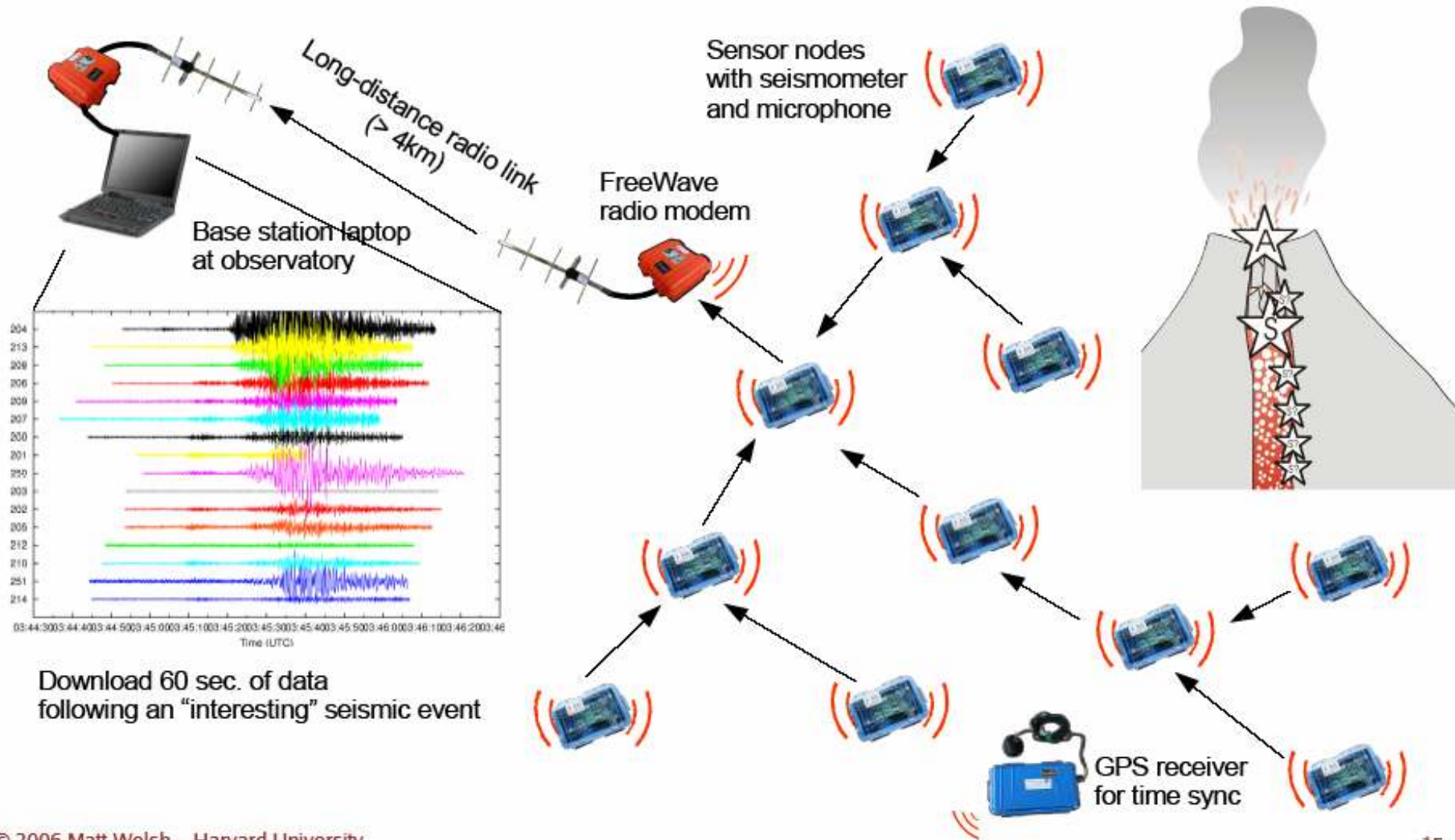
(Sámi Network Connectivity Project)

Photo: <http://news.bbc.co.uk/1/hi/technology/2491501.stm>



Preventive Maintenance on an Oil Tanker in the North Sea

Volcano System Design



Agrosens



A network diagram showing six nodes, each represented by a small square with a vertical antenna. The nodes are interconnected by thin lines, forming a network. Each node has several concentric circles around it, representing its communication range. The text "WSN Unique Characteristics" is centered in the middle of the diagram in a dark blue font.

WSN Unique Characteristics

Deployment

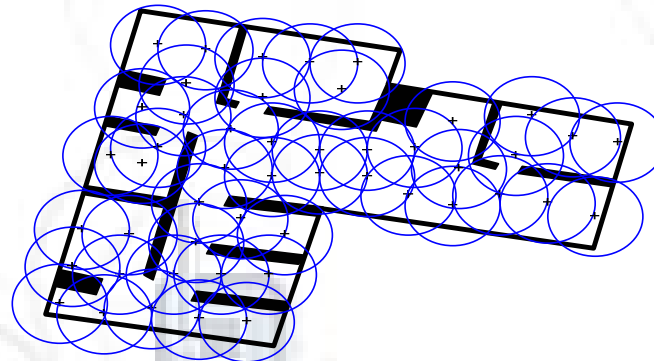
- First step in the network lifecycle
- It takes several forms:
 - Random
 - e.g. dropping them from an aircraft
 - Installed at chosen spots
 - Manual deployment is impossible in many cases
- One time vs. continuous process

Deployment



Deployment

- Problems caused by random deployment
 - Localization
 - Sensor nodes must discover their locations after deployment
 - Coverage
 - For sensing quality, a level of sensing coverage should be achieved



Coverage

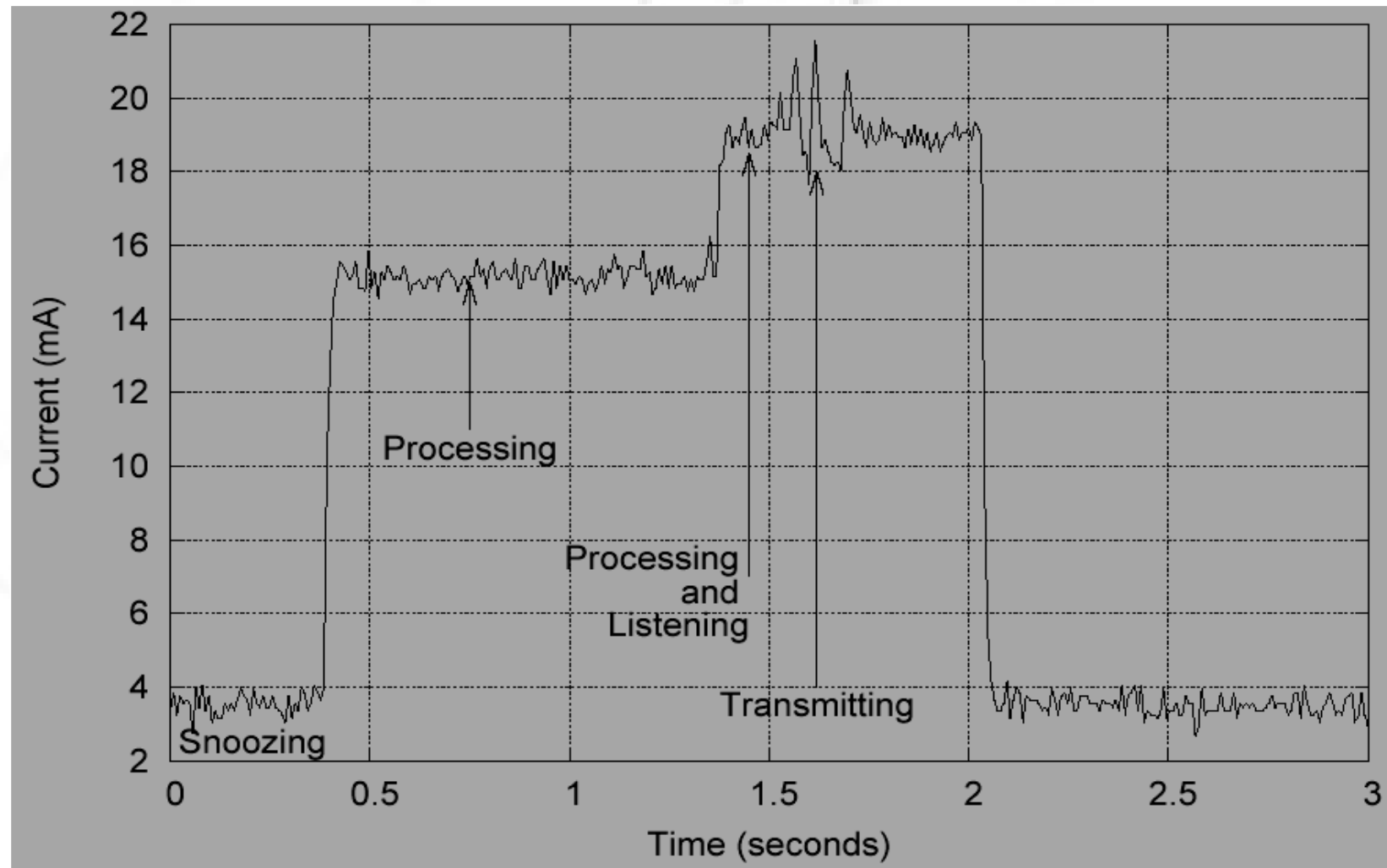
- Security
 - It is hard to store various encryption keys on nodes, since the neighborhood cannot be known a priori

Energy, Life time

- The primary limiting factor for the life time is the Energy
- Sensor nodes are driven by **batteries** and have very low energy resources
- Energy optimization** is a dominant consideration no matter what the problem is
- Radio consumes a vast majority of the system energy (transmission)

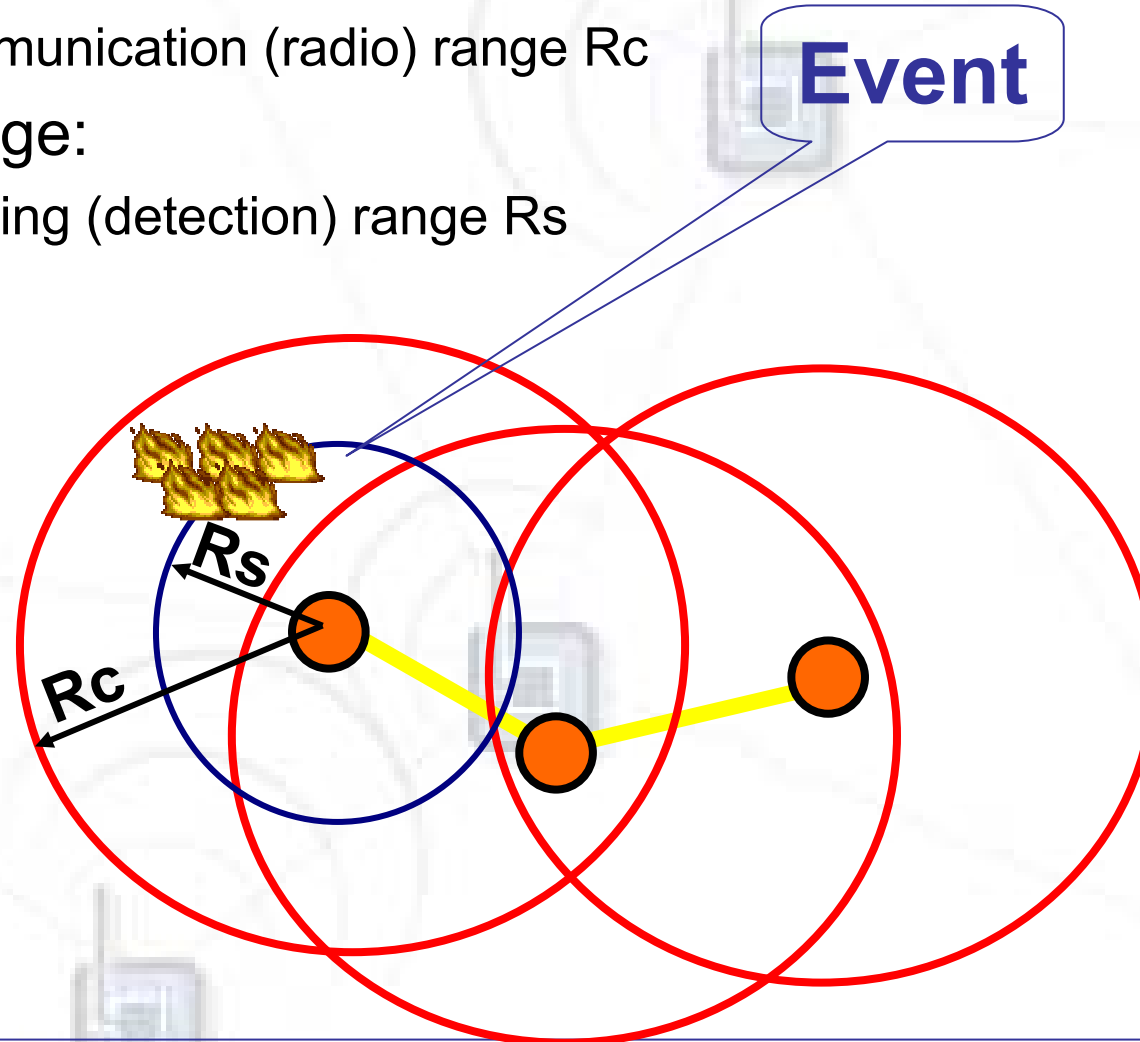
Energy, Life time

States of a sensor node



Coverage and Connectivity

- Connectivity:
 - Communication (radio) range R_c
- Coverage:
 - Sensing (detection) range R_s

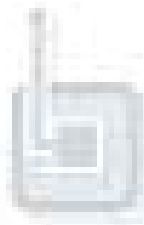


Some Challenges of WSN

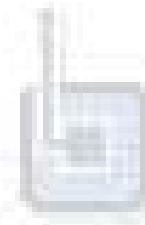
- Dynamic topology
- Clustering
- Mobility
- Multihop communication
- Periodic or event driven
- Time synchronization
- Scalability
- etc

Conclusion I

- What is wireless sensor network (WSN)?
- WSNs will enable a myriad of applications in different sectors
- Sensor networks have some unique characteristics that make the development of applications non-trivial
- Fundamental issues which the sensor network runs down
- Can we just **wait 10-15 years** until we have sensor networks that are very powerful??



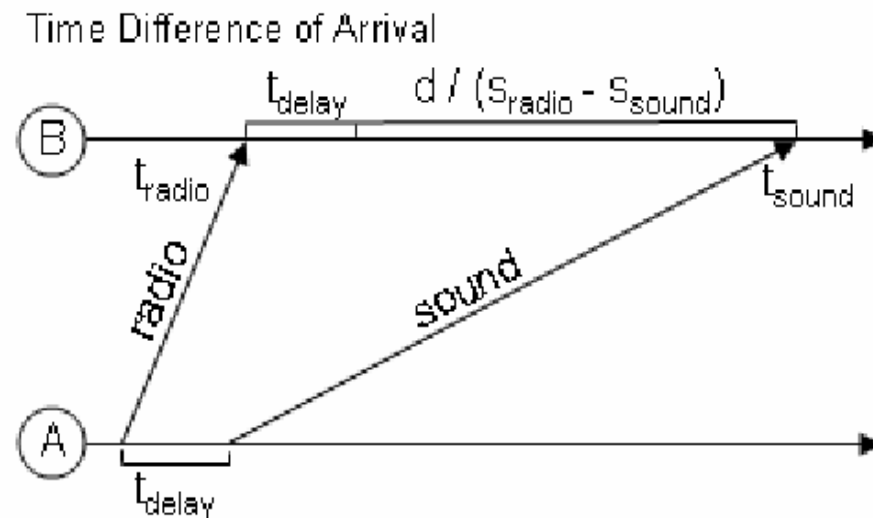
II. Localization



- Methods of computing distance measurements between nodes (Ranging)
- Is positioning necessary?
- Problem Statement
- Localization Algorithms
- Future Works
- Conclusion II

Methods of computing Distance measurements between nodes (Ranging)

- Received Signal Strength Indication (RSSI)
 - The node uses the strength of the received signal to calculate its distance from the transmitter
 - Accuracy: several meters
- Time Difference of Arrival (TDoA)
 - $d = (S_{\text{radio}} - S_{\text{sound}}) \times (t_{\text{sound}} - t_{\text{radio}} - t_{\text{delay}})$



- Angle of Arrival (AoA)

Is Positioning necessary?

- **Yes!**

- “ **Where?** ” is the question that immediately follows the detection of the event
- It can assist in geographic routing in large scale networks
- It is used to study the coverage properties of the network
- It is the mean of the Position tracking (e.g. military applications)
- etc

Problem statement

- Regular assumptions :
 - Large number of nodes
 - Random deployment in a (known shape) given area
 - Known (identical) transmission range for all nodes
 - A small number of *beacon nodes*
 - Know their locations by using GPS or being set manually
 - Static/not very dynamic networks
- Question:
 - What are the geographical positions of the nodes?

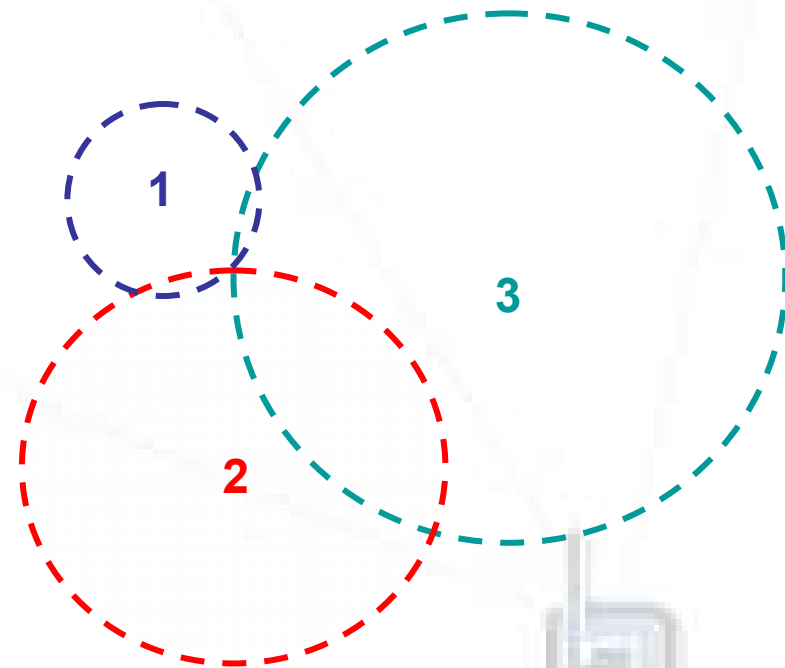
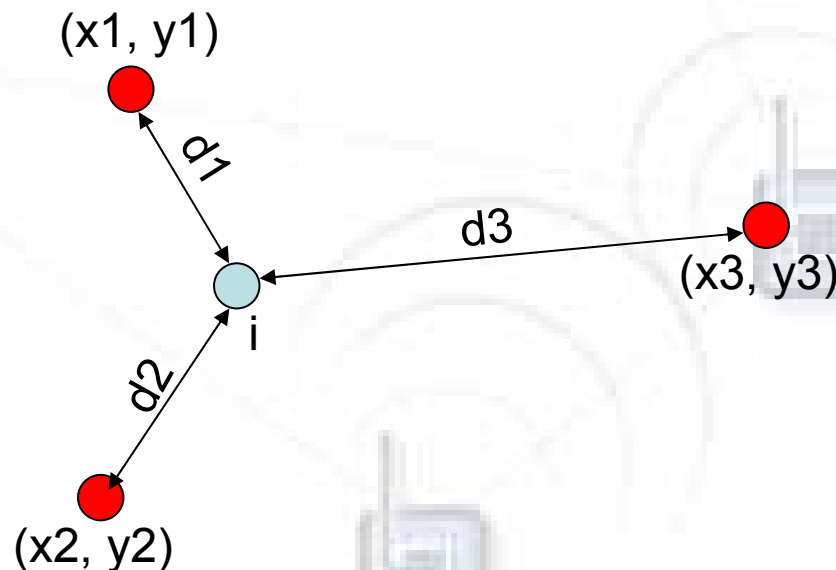
A possible solution?

- Usage of Global Positioning System (GPS) devices
- Not a feasible solution for WSN:
 - High cost of the device (value/energy/computation power /space)
 - Unavailability/poor precision of the service in special environments (indoors, underground, etc.)

Other approaches need to be developed

Lateration : a simple solution

- Given:
 - Three points with known positions
 - Distances to all three of them
- Position can be determined by intersecting three circles centered in the points with radiuses the known distances



Lateration

- The concept can be easily applied to multihop networks
- The method as such is not too useful:
 - Imprecise position information
 - Imprecise distance estimates
 - The three circles usually do not intersect in a point

Several algorithms developed on this simple idea

Localization methods (classification)

- **Centralized:**
 - All the data is collected at a central point and a global map is computed at once
- **Distributed:**
 - nodes compute their position by communicating to their neighbours only

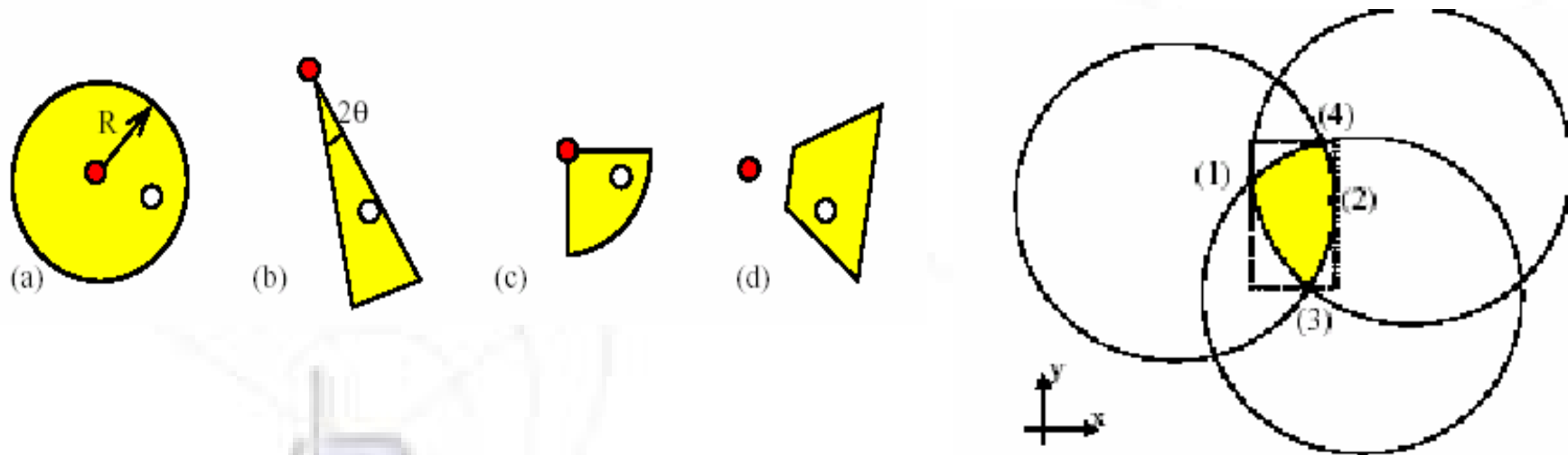
Centralized	Distributed
High costs of Communication	Lower Communication overhead
High quality solution (average error)	The resulting positions are less precise
High computation power required	Low complex methods
Global maps available	High number of anchors needed



Centralized Methods

Convex Optimization (1/2)

- One of the first schemes available
- Treats the localization problem from the point of view of linear programming and semi-definite programming
- Various constraints are represented as linear matrix inequalities



Convex Optimization (2/2)

- **Advantages:**

- Elegant algorithm (given a set of convex constraints on a node's position, SDP simply finds the intersection of the constraints)
- Efficient computational methods have already been developed

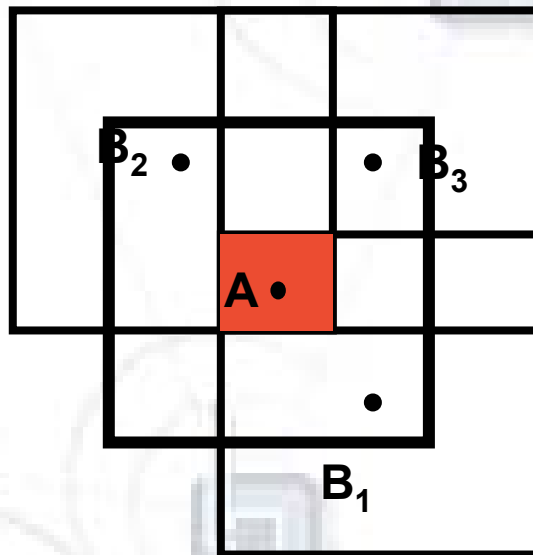
- **Disadvantages:**

- All the disadvantages of the centralized methods
- Computation complexities:
 - Linear programming is quadratic in the number of connections
 - Semi-definite is cubic in the number of connections



Distributed Methods

Bounding Box (1/2)



Step 21: In the second stage, the pseudo-beacon node is identified at the intersection of the bounding boxes of the three beacon nodes. A node which hears all three beacon nodes is termed as a pseudo-beacon node



Bounding Box ^(2/2)



- **Disadvantages:**
 - All the disadvantages of the distributed methods
 - Accuracy few meters
- **Advantage:**
 - Bounding box works best when sensor nodes have extreme computational limitations, since other algorithms may simply be infeasible



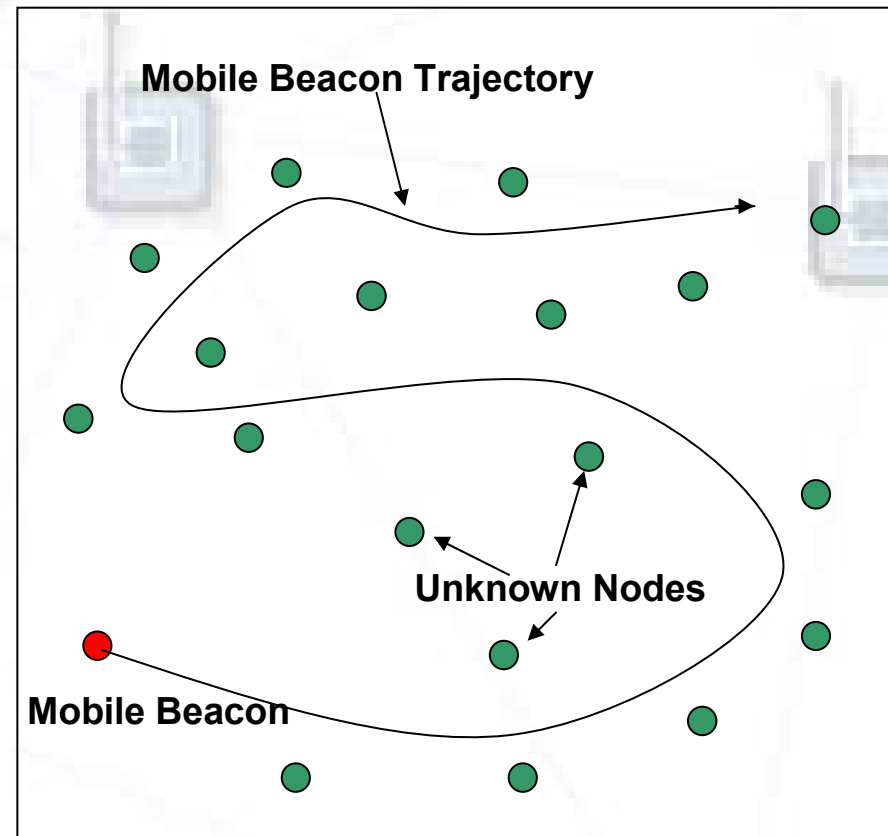
Localization with a mobile beacon

Localization with a mobile beacon

- **Static Beacons Based Localization Approaches:**
 - The beacons positions are unchangeable
 - Disadvantages:
 - The precision increases with the number of beacons, which are expensive
 - May incur large errors in estimation and measurements if a node is far from the group
- **Mobile Beacon Based Localization Approaches :**
 - The beacon propagates in the region of monitoring and communicates with the unknown nodes
 - Main Drawback:
 - The lack of a well defined mobile beacon trajectory

Localization with a Mobile Beacon

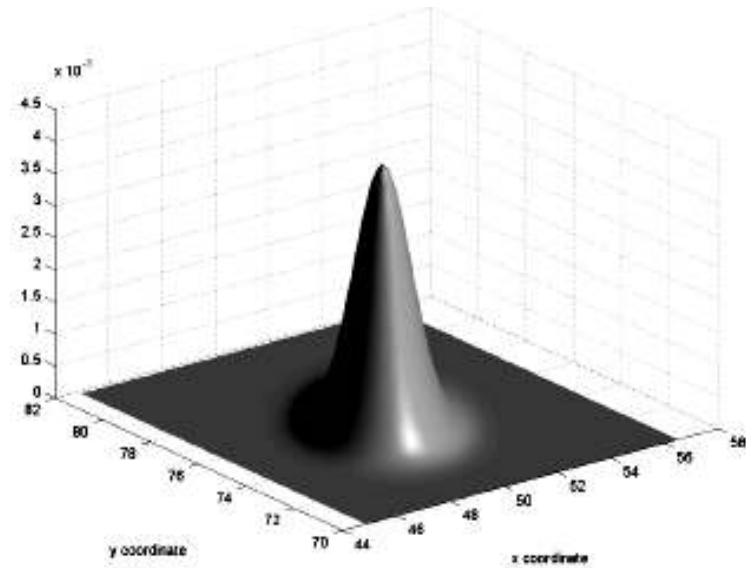
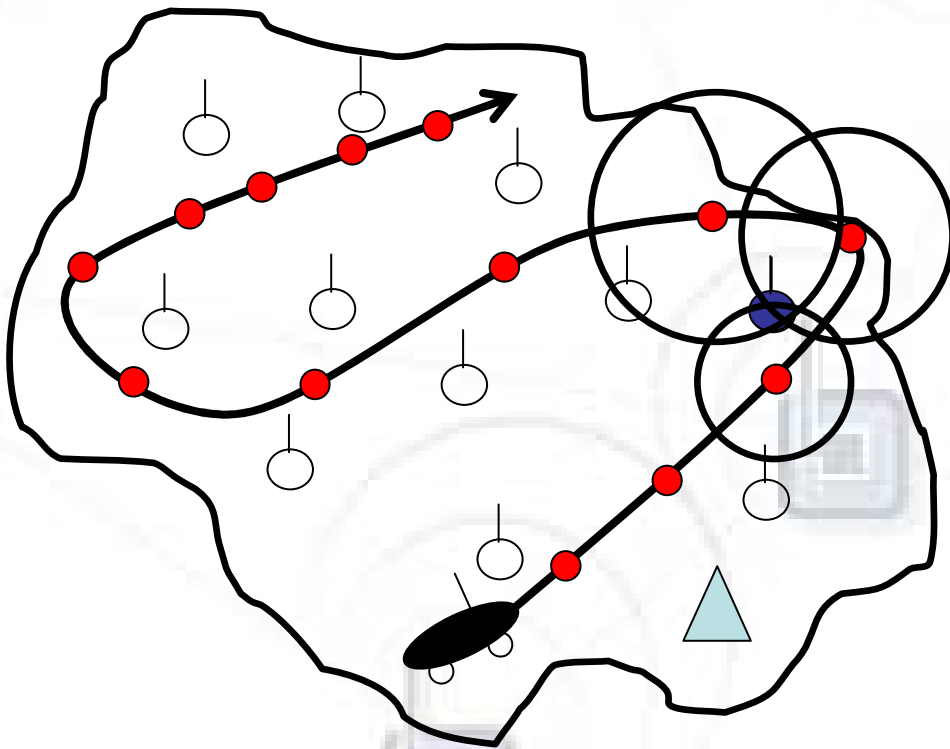
- One mobile beacon (equipped with GPS, etc)
- After deployment, the mobile beacon traverses the sensor network while broadcasting beacon packets
- The beacon packet contains the coordinates of the beacon
- Any node receiving the packets, will be able to locate itself through localization algorithms (ex: trilateration)



What is the optimum beacon trajectory and when should the beacon packets be sent?

Localisation à l'aide d'une ancre mobile

- Une seule ancre mobile au lieu de plusieurs fixes



Problématique ouverte !

Les inconvénients d'une telle approche :

1. Absence d'une trajectoire bien définie

Courbe de Hilbert

2. Les instants de diffusion de l'information et son contenu

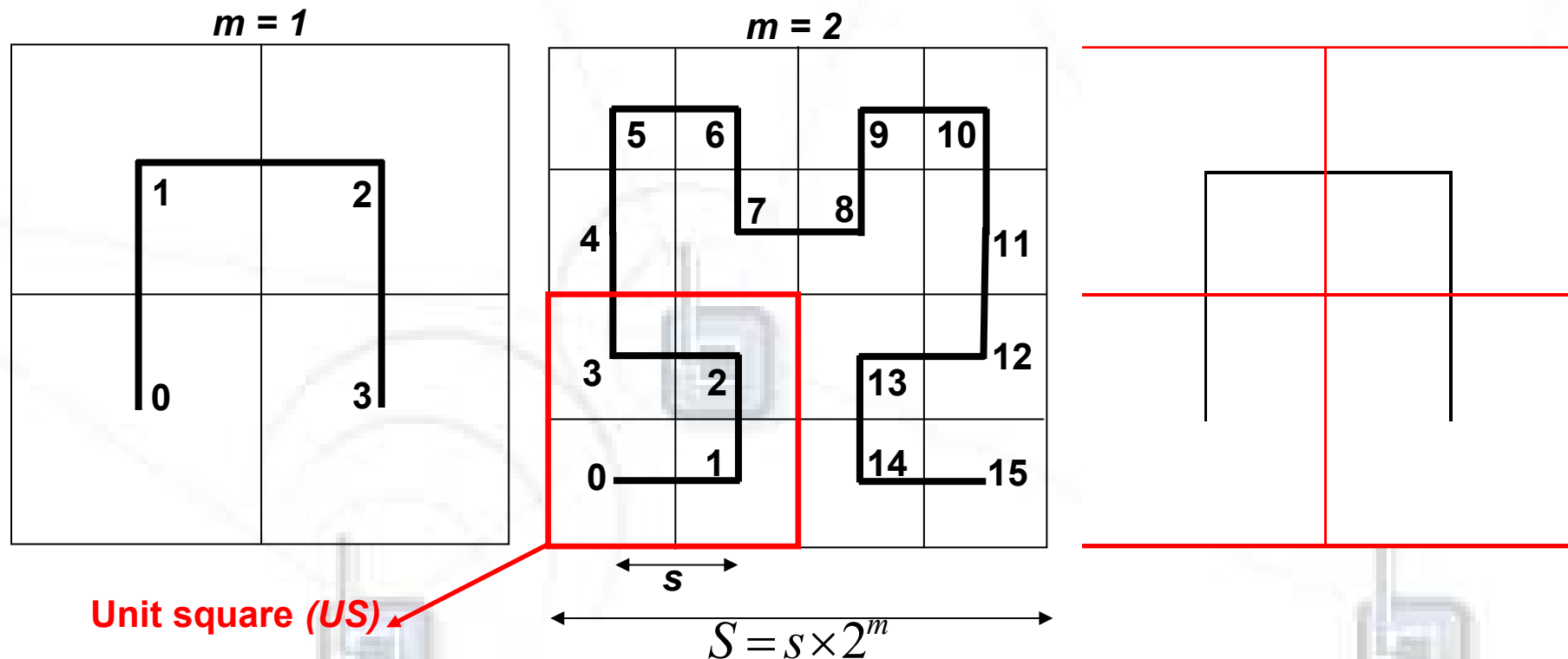
Les clefs-Hilbert

3. La complexité $O(n^2)$

Deux algorithmes de faible complexité $O(1)$

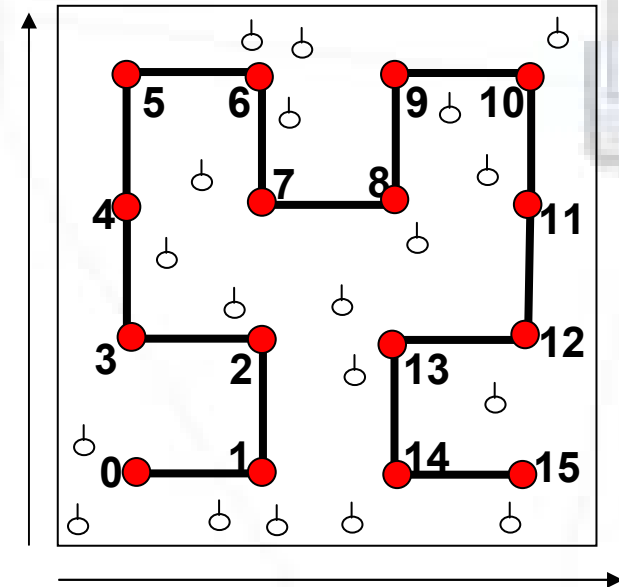
Trajectoire de l'ancre mobile

- Rappel : Courbe de Hilbert
 - Courbe remplissant l'espace
 - Possédant un Ordre noté « m »
 - Clefs-Hilbert



Algorithmes de localisation

- Zone parcourue selon une trajectoire de Hilbert
- Diffusion des h-clefs
- 3 h-clefs nécessaires pour la localisation



Lemme : Un nœud n appartenant à un carré unité US , reçoit au moins 3 h-clefs envoyées dans son US si et seulement si $s \leq \sqrt{2/5} \times Rc$

Par conséquent nous obtenons : $m \geq \lceil \log_2 (S / \sqrt{2/5} \times Rc) \rceil$

Algorithmes de localisation

Algorithme de Hillbert :

- Étape 1 : Calcul de la position (X_n, Y_n) du nœud envoyé dans son US
- Étape 2 : $h_3(X_2, Y_2)$ et $h_2(X_2, Y_2)$ permettent le calcul de α sachant que h_3 et h_2 avaient été envoyées

$$\cos(\alpha) = \frac{(a_2 + s - a_3)^2 + s^2 - d_2^2}{2 \times s \times d_2}$$

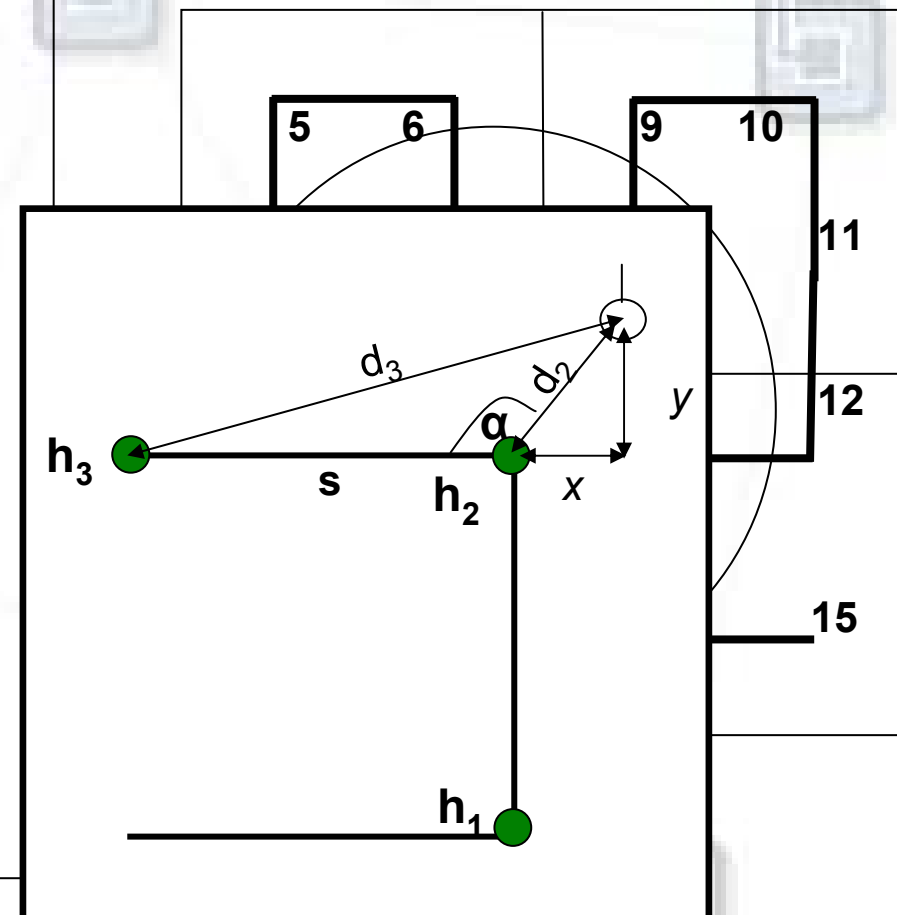
- Utilisation de RSSI ou TDoA

$$\cos(\alpha) < 0 \text{ et } X_2 > X_3 \Rightarrow$$

$$X_n = X_2 + x$$

- De la même manière en utilisant h_1 et h_2 nous trouvons:

$$Y_n = Y_2 + y$$

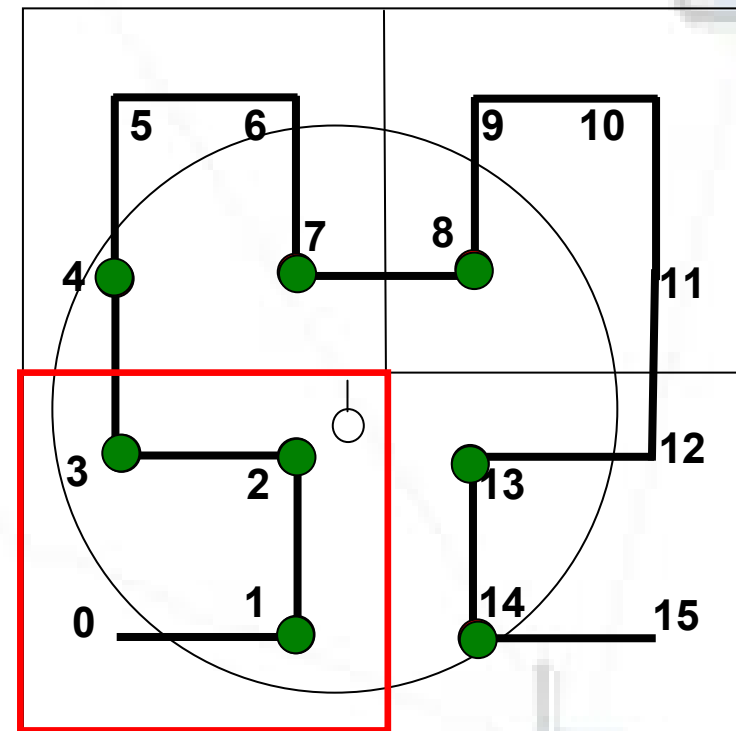


Algorithmes de localisation

- Algorithme bary-Hilbert :
 - Sélection des triplets de h-clefs formant une courbe « L »
 - Pour chaque triplet
 - exécution de l'algorithme-Hilbert
 - estimation de position (X_{ni}, Y_{ni})
 - Position du nœud :

$$X_n = \sum_{i=1}^l X_{ni} / l \quad \text{et} \quad Y_n = \sum_{i=1}^l Y_{ni} / l$$

"l" est le nombre de courbes en «L» sélectionnées



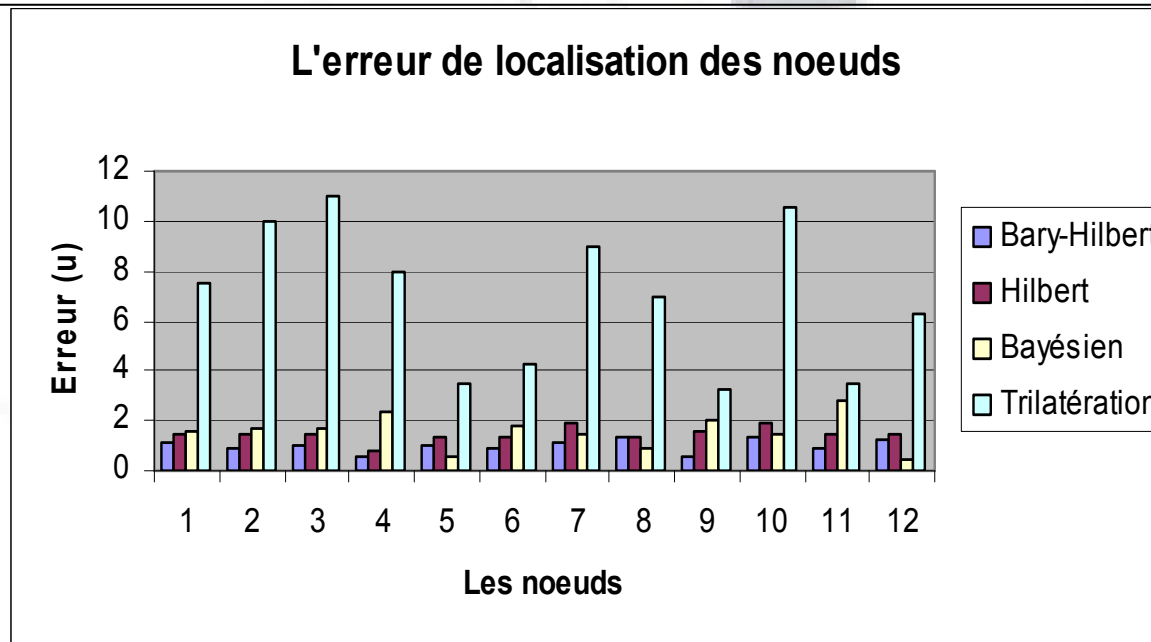


Évaluation

- Simulateur : OMNet ++
- Paramètres :
 - Le réseau
 - Surface
 - Déploiement
 - Nombre de nœuds
 - L'ancre mobile
 - Variation du rayon de communication R_c
 - L'estimation de la distance
 - RSSI et TDoA

Performance

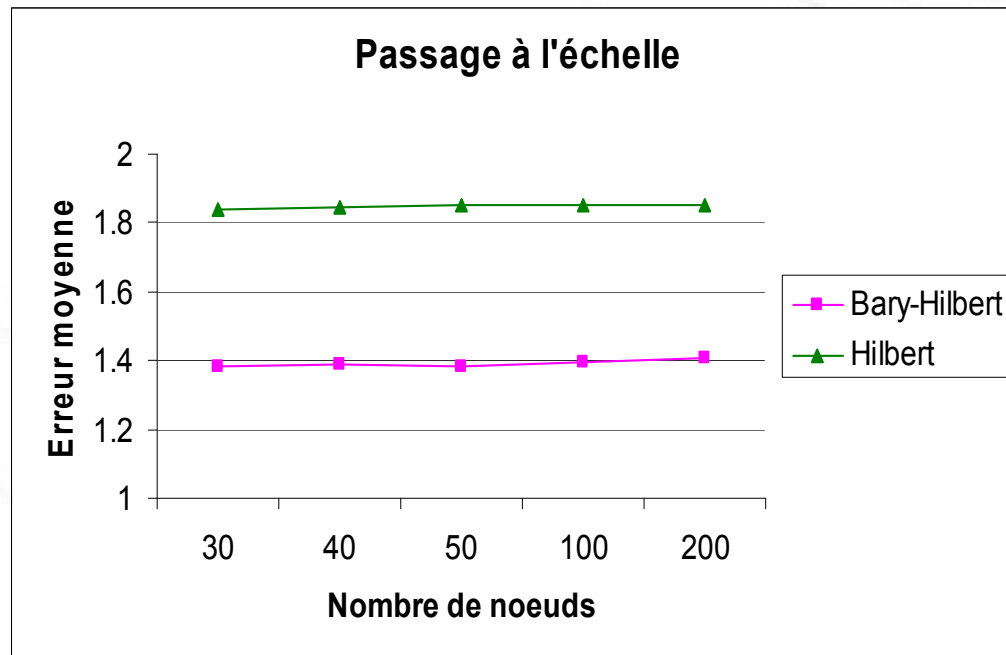
- **Comparaison :**
 - Algorithme de trilatération, Algorithme Bayésien, Hilbert et bary-Hilbert
 - 12 nœuds déployés dans une surface de 17×65 us



Algorithme	Bary-Hilbert	Hilbert	Bayésien	trilatération
Complexité	O(1)	O(1)	O(n²)	O(1)
Erreur moyenne	1,1	1,3	1,4	9,8

Performance

- **Passage à l'échelle**



- **Rayon de communication (200 nœuds)**

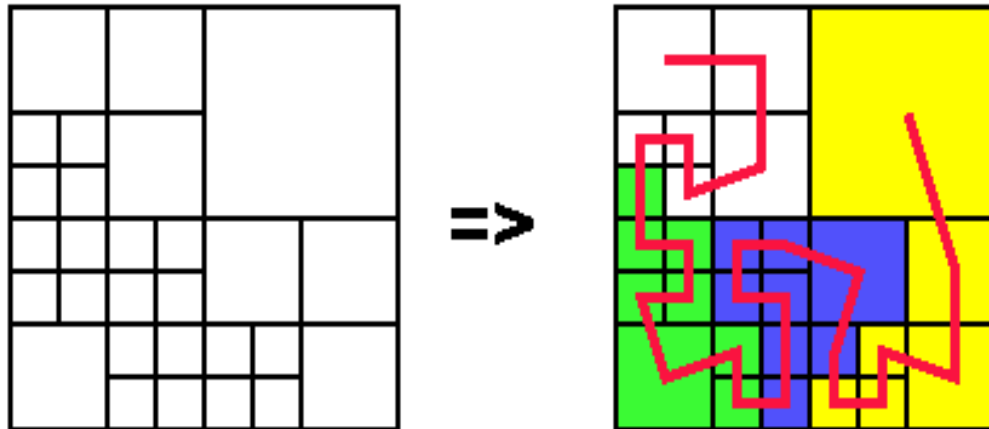
Rc	« m »	Erreur Hilbert	Erreur Bary-Hilbert
10	4	1,15	0,97
15	3	1,84	1,39
20	3	2,18	1,78
25	3	2,79	2,1
30	2	3,11	2,64

Conclusion II

- Les réseaux de capteurs : une technologie récente
 - Contraintes : consommation d'énergie, topologie dynamique, etc.
 - Très orientés vers un domaine d'application précis
 - Problématiques traitées : localisation, couverture, fusion de données
- Localisation :
 - Approche basée sur ancre mobile
 - Choix de la trajectoire de Hilbert
 - Deux algorithmes extensibles et à faible complexité
 - **Limitations : ancre mobile et communications**

Perspectives

- Changement de l'ordre de la courbe de Hilbert en temps réel
 - Selon des critères tel que la densité ou la précision définie



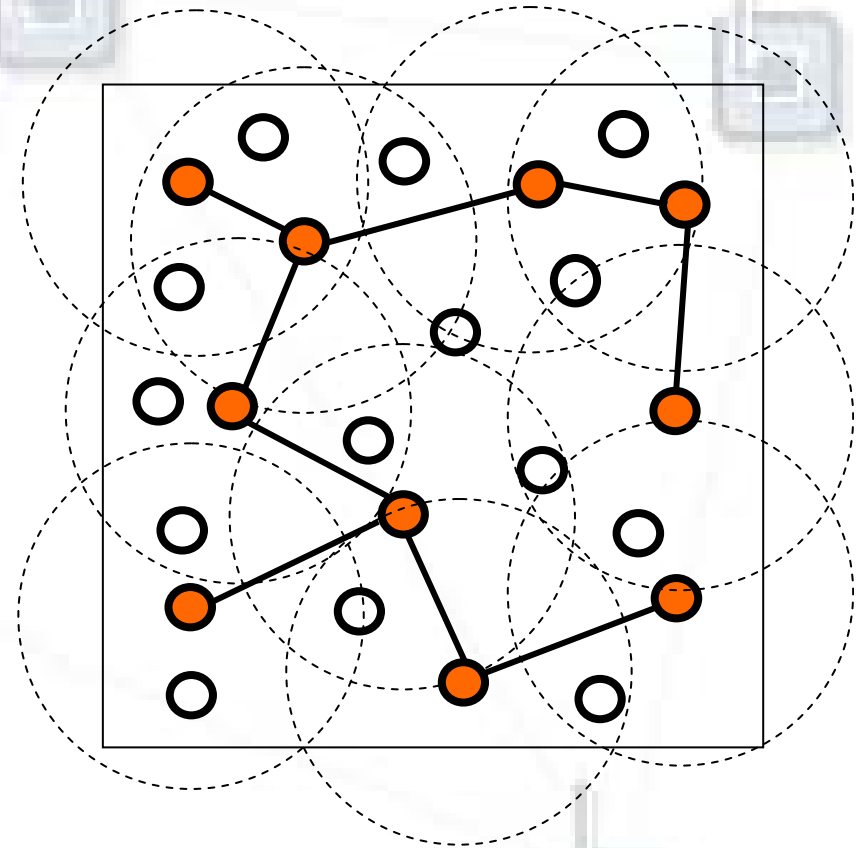
- Le problème de la couverture
- Utilisation de la courbe de Hilbert pour le calcul de la moyenne des mesures d'une manière sérialisée

Deuxième partie

Couverture et maximisation de la durée de vie
du réseau

Couverture et optimisation d'énergie

- Couverture : qualité de détection
- Nœuds redondants en mode veille
- Ordonnancement des nœuds dans des sous-ensembles d'activation disjoints couvrant et connexe
- À un instant « t » un et un seul ensemble actif
- Round : durée de vie d'un ensemble actif de nœuds



Ordonnancement centralisé et distribué

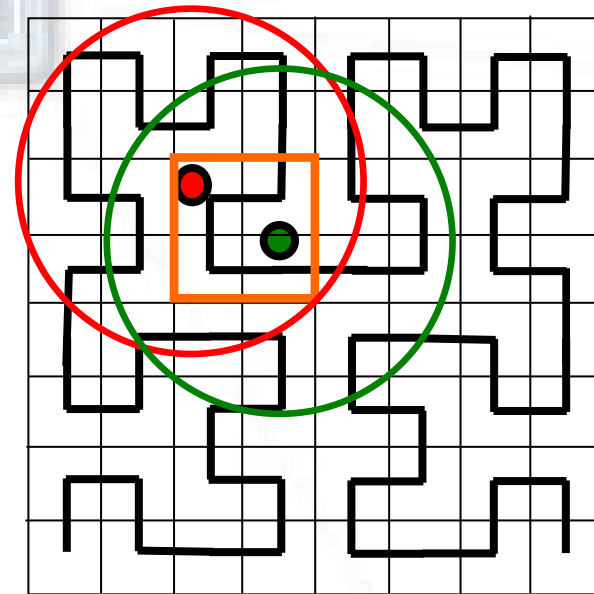
- Une approche centralisée pour construire des ensembles dominants disjoints est présentée par *Cardei et al* :
 - Le réseau est modélisé par un graphe non orienté de n nœuds
 - Deux nœuds sont redondants s'ils se trouvent dans la zone de couverture l'un de l'autre
 - Problème NP-complet => Heuristique basée sur la méthode de coloration des graphes ($O(n^3)$)
- Une approche distribuée, présentée par *Carle et al*, consiste à trouver les « connected area dominating sets » :
 - Découverte de voisins
 - Sélection des relais adaptés à la couverture
 - Décision d'activation

Notre approche : ordonnancement par une ancre mobile

- Coupler la localisation et la couverture en une seule phase
- Éviter la phase de découverte de voisins
- Diminuer de manière significative la consommation d'énergie

Couverture avec l'ancre-Hilbert

- Impact de la couverture sur l'ordre de la trajectoire
 - Un nœud se trouvant dans un US le couvre complètement
- Nœuds appartenant au même carré unité US : redondants
- Trois algorithmes d'ordonnancement



Lemme : Un nœud n appartenant à un carré unité US le couvre entièrement si et seulement si $s \leq R_s / 2\sqrt{2}$

Par conséquent nous obtenons : $m \geq \left\lceil \log_2 \left(2\sqrt{2}S / R_s \right) \right\rceil \geq \left\lceil \log_2 \left(S / \sqrt{2/5} \times R_c \right) \right\rceil$

Couverture incrémentale

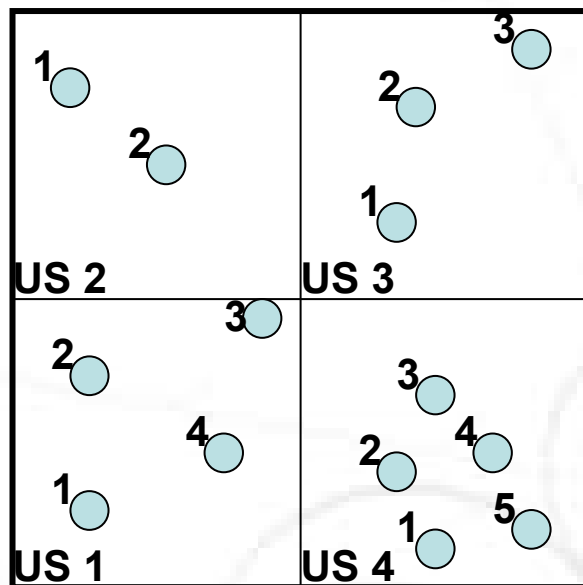
Ordonnancement aléatoire :

- $R_c \geq 2 \times R_s$ (condition de *Zhang et Hou*)
- Après sa localisation, un nœud envoie sa position à l'ancre mobile (une seule communication)
- L'ancre reçoit les positions qui se situent dans le même *US* et elle les considère comme voisins redondants
- Elle affecte chaque nœud à un ensemble (round)
- À la fin, elle renvoie sa décision aux nœuds en précisant à chaque nœud son ensemble

Couverture incrémentale

Ordonnancement aléatoire

– Exemple :



	US 1	US 2	US 3	US 4
Round 1	n_{11}	n_{21}	n_{31}	n_{41}
Round 2	n_{12}	n_{22}	n_{32}	n_{42}
Round 3	n_{13}	-	n_{33}	n_{43}
Round 4	n_{14}	-	-	n_{44}

- Ordonnancement à faible complexité => durée de vie du réseau améliorée
- Optimalité?

Couverture incrémentale

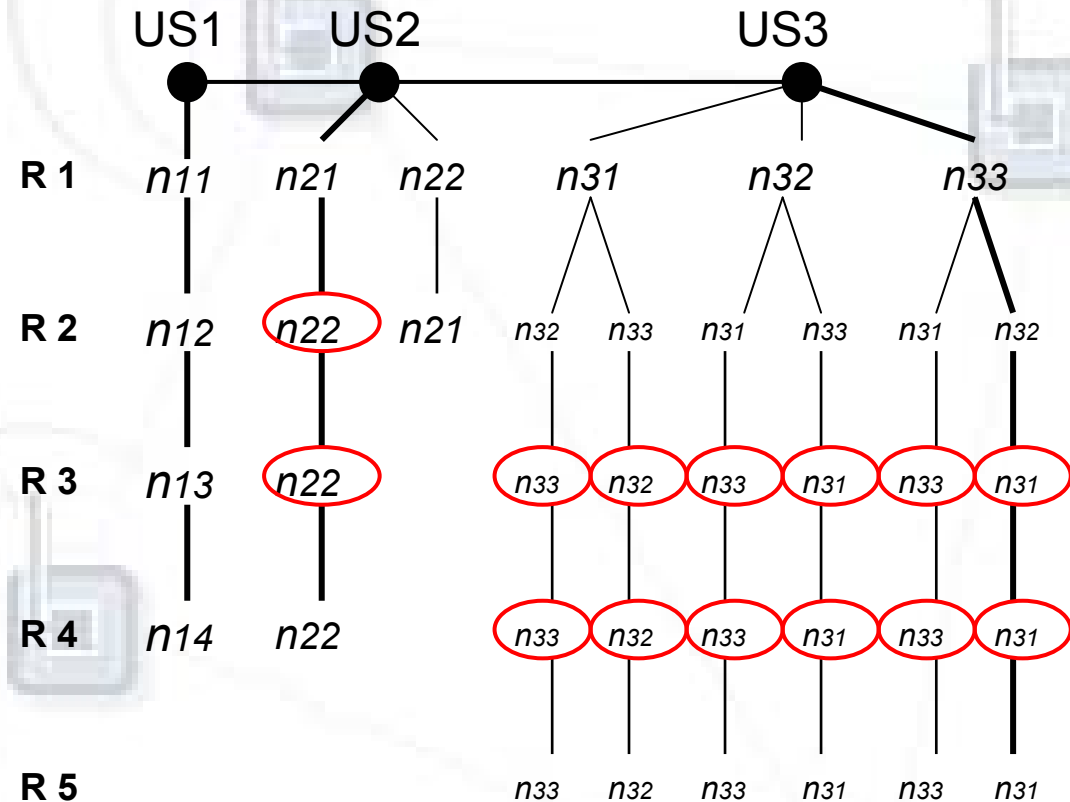
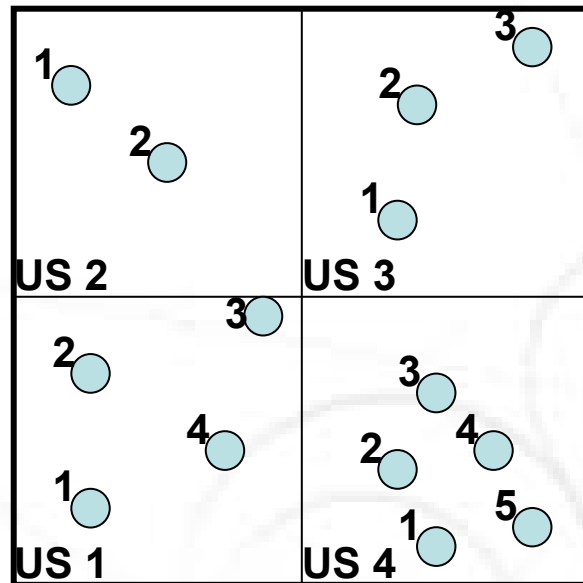
Ordonnancement global :

1. Tous les ordonnancements possibles sont construits en tenant compte de la connectivité
2. Parmi les ordonnancements construits, celui qui assure une couverture complète sera sélectionné (Monte-Carlo)

Couverture incrémentale

Ordonnancement global

- Branch & Bound (exemple)



- Solution optimale mais complexe ($O(n!)$)!!

Couverture incrémentale

Ordonnancement par la distance maximale :

- Réduction du coût de la recherche exhaustive et du calcul de la surface couverte ($O(n \log(n))$)
- Idée : réduction de la surface de chevauchement (surface couverte par plusieurs nœuds)
 - Le chevauchement entre deux disques diminue avec l'augmentation de distance entre leurs centres
- Le même principe est appliqué à deux ensembles de nœuds au lieu de deux disques
- La distance maximale entre deux ensembles de points coplanaires est définie par :

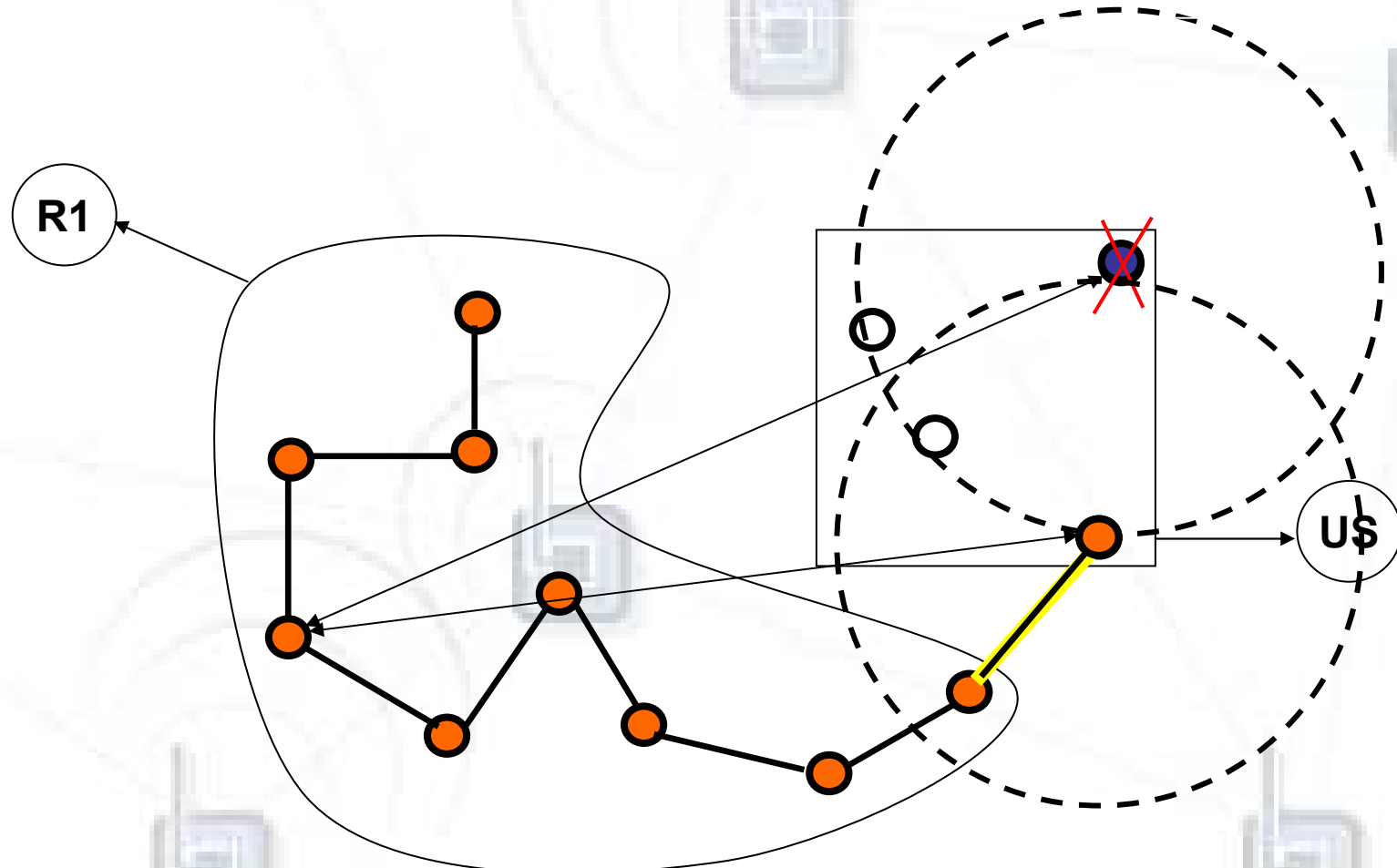
$$d_{\max}(S_1, S_2) = \max_i^j d(p_i, q_j)$$

où $d(p_i, q_j)$ est la distance euclidienne entre les points p_i de S_1 et q_j de S_2

Couverture incrémentale

Ordonnancement par la distance maximale :

- Exemple :



Évaluation

- Séries de simulation basées sur OMNet++
- Génération du réseau d'une manière aléatoire dans une région (50×50) us, $R_s = 8$ u, $m = 4$ u, $s = 2.82$ u
- Résultats = moyenne d'une cinquantaine de simulations
- Implémentation de trois ordonnancements notamment: global, aléatoire, distance maximale
- Ordonnement global : très forte complexité ($O(n!)$) et temps d'exécution exponentiel (par exemple 30 minutes pour une densité de 50 nœuds)

A diagram of a wireless sensor network is visible in the background. It consists of several nodes, each represented by a small square with a vertical antenna on top. These nodes are interconnected by thin lines, forming a network topology. Some nodes have concentric circles around them, representing their communication range or coverage area. The nodes are distributed across the slide, with some at the top, some in the middle, and some at the bottom.

Évaluation

- Les principaux paramètres mesurés :
 - Durée de vie du réseau
 - Proportion des nœuds actifs par rapport au nombre total déployé
 - Pourcentage de couverture
 - Pourcentage de chevauchement

Comparaison

- Approche centralisée (coloration de graphe)
 - Topologie : 100 nœuds, surface (100×100) us

Algorithme	Centralisé	Aléatoire	Distance-max
Nb de rounds	5	6	8

- Approche distribuée (CADS)
 - Topologie : 70 nœuds

Algorithme	Distribué	Aléatoire	Distance-max
Nb de rounds	2.7	3	4

Pourcentage de couverture

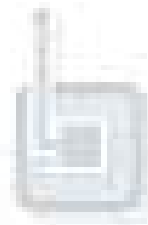
	<i>Ordonnancement aléatoire</i>			
	R1	R2	R3	R4
D-30	100%	70%	DN	DN
D-50	100%	81%	DN	DN
D-70	100%	91%	67%	DN
D-100	100%	93%	75%	DN
D-150	100%	100%	96%	85%

	<i>Ordonnancement par la distance maximale</i>						
	R1	R2	R3	R4	R5	R6	R7
D-30	100%	73%	DN	DN	DN	DN	DN
D-50	100%	93%	61%	DN	DN	DN	DN
D-70	100%	93%	81%	64%	DN	DN	DN
D-100	100%	96%	84%	79%	75%	DN	DN
D-150	100%	100%	93%	87%	85%	75%	70%

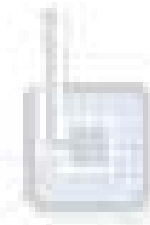
Pourcentage de chevauchement

	Chev. Initial	<i>Ordonnancement aléatoire</i>			
		R1	R2	R3	R4
D-30	79%	50%	5%		
D-50	96%	75%	26%		
D-70	100%	80%	35%	3,5%	
D-100	100%	85%	45%	30%	
D-150	100%	95%	77%	40%	26%

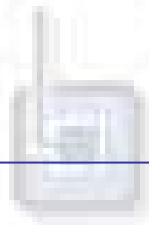
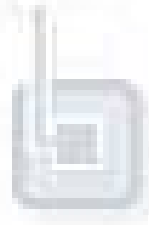
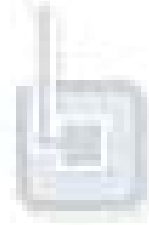
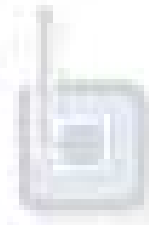
	Chev. Initial	<i>Ordonnancement par la distance maximale</i>					
		R1	R2	R3	R4	R5	R6
D-30	79%	45%	5%				
D-50	96%	30%	23%	15%			
D-70	100%	38%	29%	20%	11%		
D-100	100%	40%	33%	20%	18%	17%	
D-150	100%	45%	41%	31%	22%	20%	10%



Troisième partie



Fusion de données



Processus de fusion de données

- Les nœuds capteurs prennent des mesures de leur environnement
- Calcul de la moyenne de tous les prélèvements => Effectuer une action concertée
 - Tâche facile dans des réseaux fiables
 - Problématique dans un réseau sujet à la fois à des perturbations environnementales et à des défaillances fréquentes

Calcul de la moyenne consensus

- Calcul centralisé :
 - Envoie des mesures vers une unité centrale
 - Inconvénients :
 - routage, topologie dynamique
 - consommation d'énergie
- Calcul distribué (flooding) :
 - Chaque nœud diffuse son information et les informations reçues à ses voisins
 - Ayant toutes les informations, il calcule la moyenne
 - Inconvénients :
 - messages doubles, réception du même message plusieurs fois
 - besoin d'une grande mémoire, ...

Calcul de la moyenne consensus

- Calcul itératif :
 - Un nœud maintient un état dynamique de l'estimation de la moyenne
 - Échange des informations
 - Réception des messages
 - Mise à jour de l'état dynamique
 - Convergence vers la moyenne consensus
- Travaux itératifs existants :
 - Tolérants aux cassures des liens (topologie dynamique)
 - Synchrones
 - Non tolérants aux pertes des messages
 - Non résistants aux délais de transmission

Algorithme itératif asynchrone

- Algorithme itératif asynchrone pour le calcul de la moyenne consensus
 - Basé sur la diffusion
 - Asynchrone
 - Robuste aux cassures des liens
 - Résiste aux délais de transmission
 - Tolérant aux pertes de messages

Algorithme itératif asynchrone

- Fusion Asynchrone :

- Chaque nœud maintient un état instantané
- Après chaque itération « t » un nœud i :

- Compare son état $x_i(t)$ avec les états de tous ses voisins $x_j^i(t) = x_j(d(t))$

- Calcule un scalaire $s_{ij}(t)$ pour tout voisin j

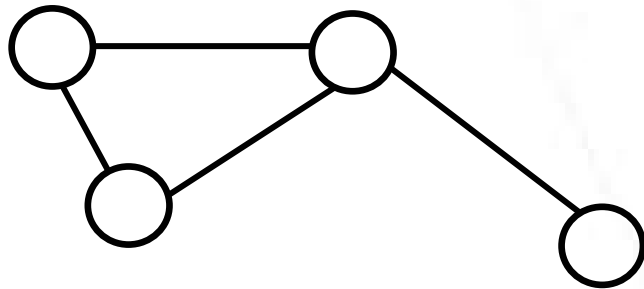
- Reçoit les scalaires envoyés par ses voisins $r_{ji}(t) = s_{ji}(d(t))$

- Met à jour son état instantané tel que :

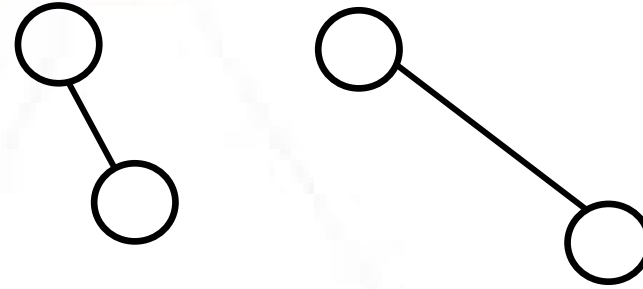
$$x_i(t+1) = x_i(t) - \sum_{j \in N_i(t)} s_{ij}(t) + \sum_{j \in N_i(t)} r_{ji}(t)$$

Algorithme itératif asynchrone

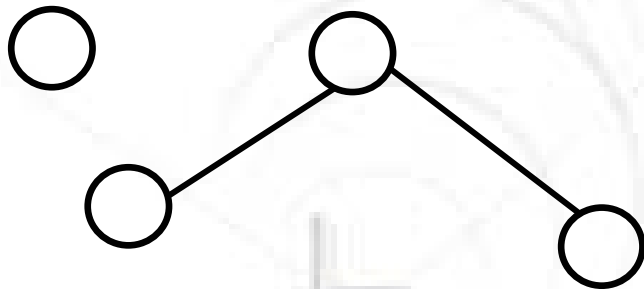
Hypothèse 1: superposition des graphes de communication forme un graphe connexe



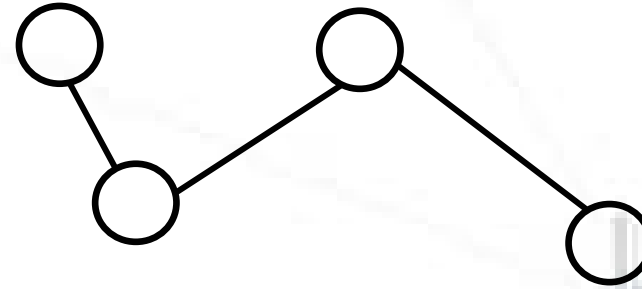
Réseau initial



Réseau à l'instant « t »



Réseau à l'instant « t+1 »



Graphe de superposition

Algorithme itératif asynchrone

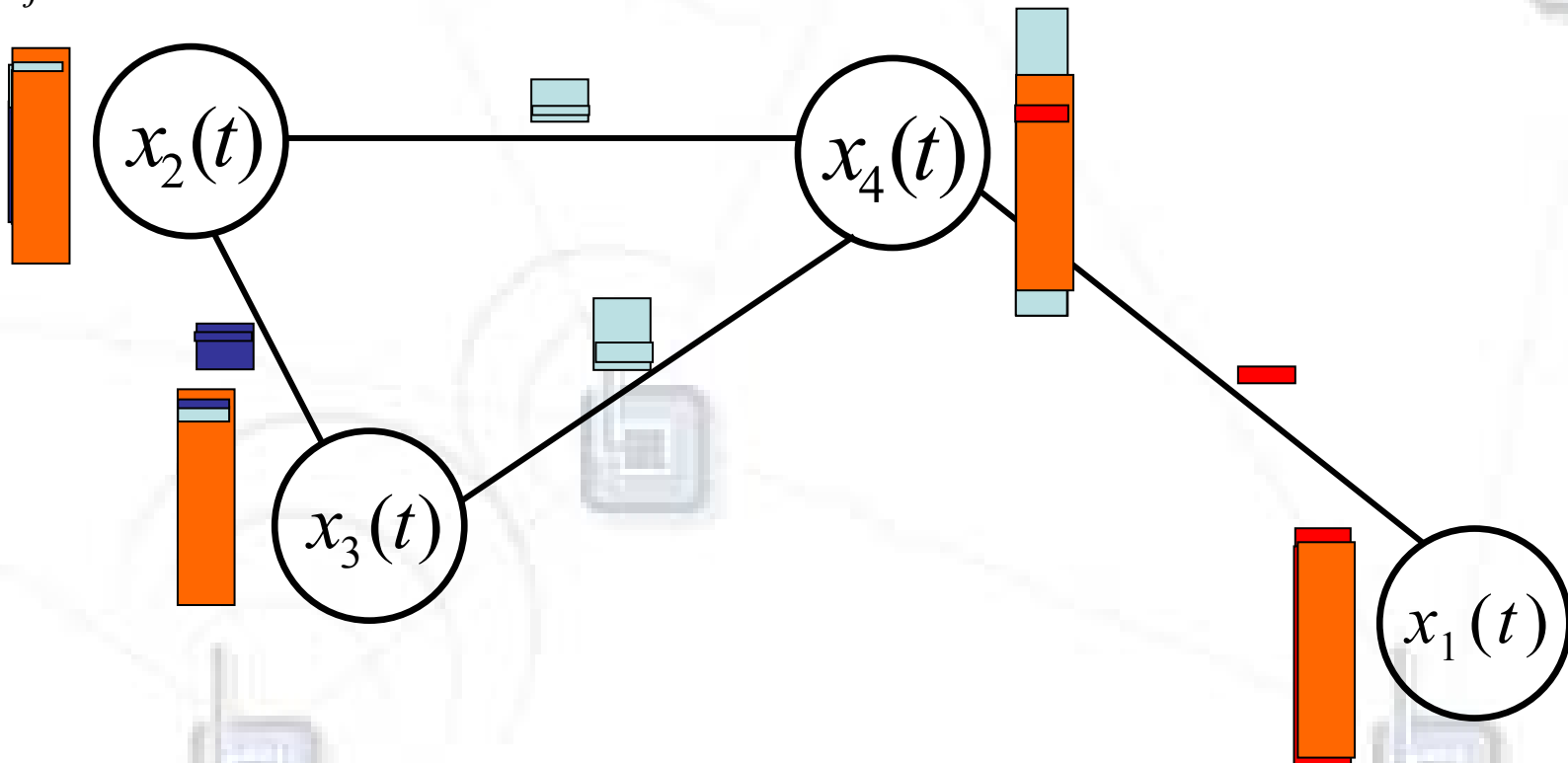
Hypothèse 2: $\exists \alpha \in [0,1], \exists j^*$ voisin du noeud i tel que $x_{j^*}^i(t) = \min x_k^i(t)$,
 $\alpha(x_i(t) - x_{j^*}^i(t)) \leq S_{ij^*}(t)$

Hypothèse 3: Condition ping pong

$$x_i(t) - \sum_{K \in N_i(t)} S_{ik}(t) \geq x_j^i(t) + S_{ij}(t)$$

Algorithme itératif asynchrone

Choix des S_{ij} : $S_{ij}(t) = \alpha_{ij}(t)(x_i(t) - x_j^i(t))$ si $x_i(t) > x_j^i(t)$
 $\alpha_{ij}(t) = 1 / (N_i(t) + 1)$ tant qu'on n'a pas ping - pong

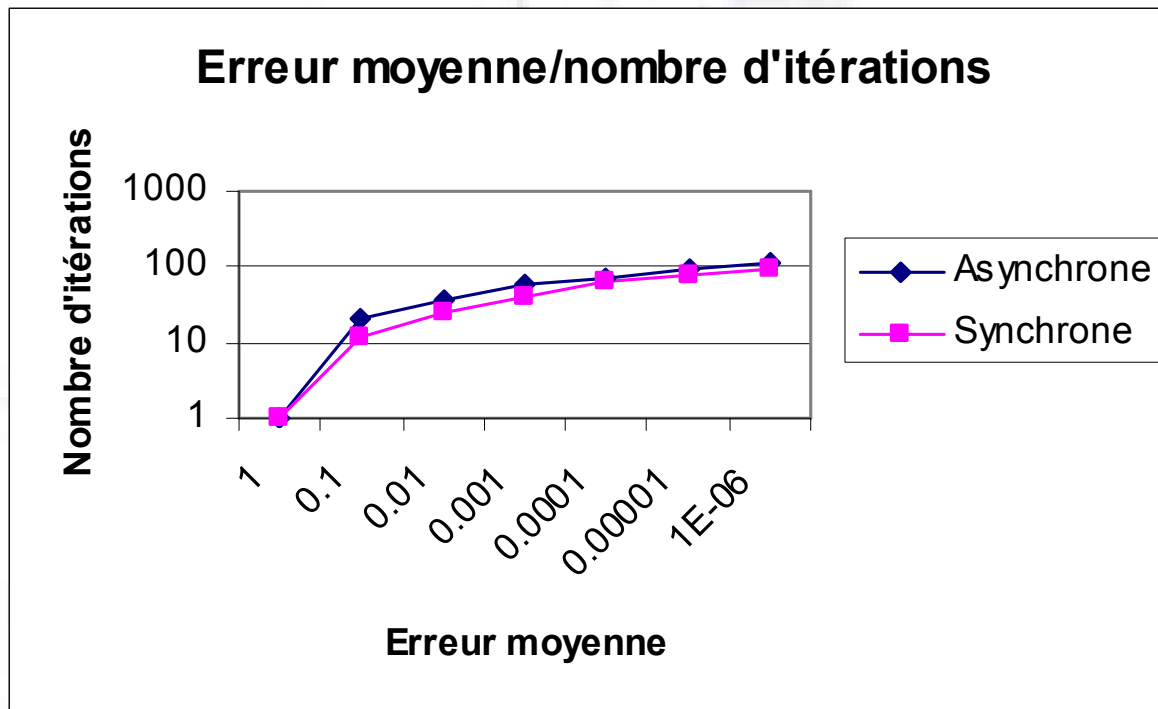


Évaluation

- Séries de simulations (50 exécutions par simulation) avec OMNet++
- Paramètres :
 - Rapport T_r = durée itération / durée communication
 - Le nombre de nœuds
 - La probabilité de cassure de lien
- Métriques évaluées :
 - Nombre d'itérations
 - Erreur moyenne
 - Temps de convergence

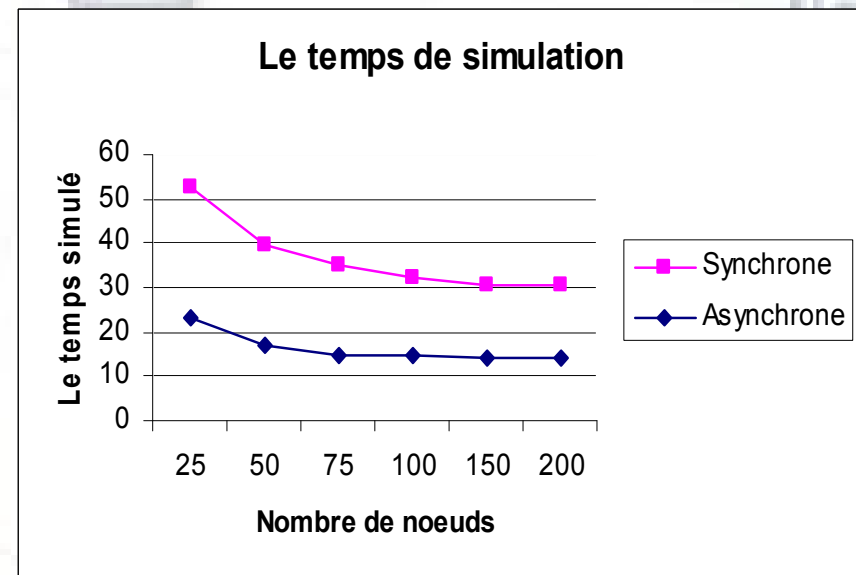
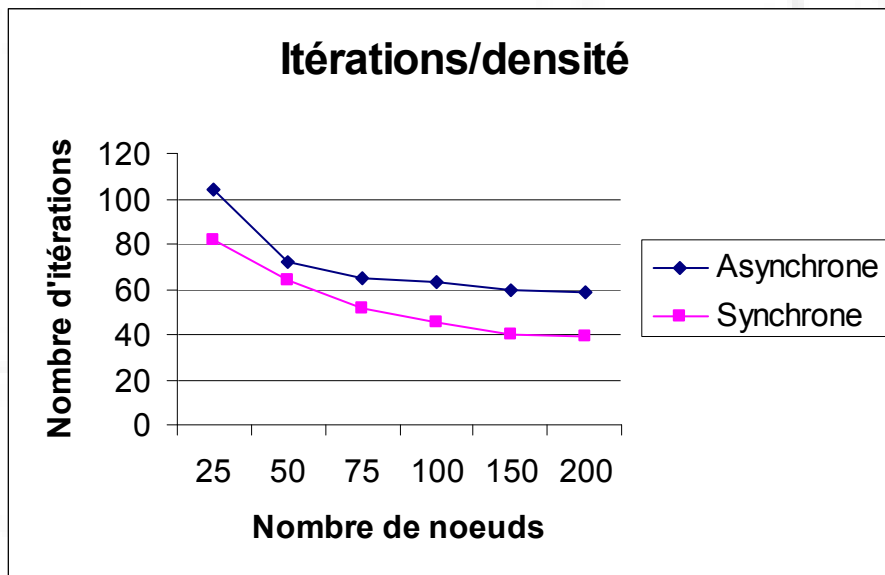
Évaluation

- Comportement :
 - Topologie fixe (Erreur moyenne, 100 nœuds)



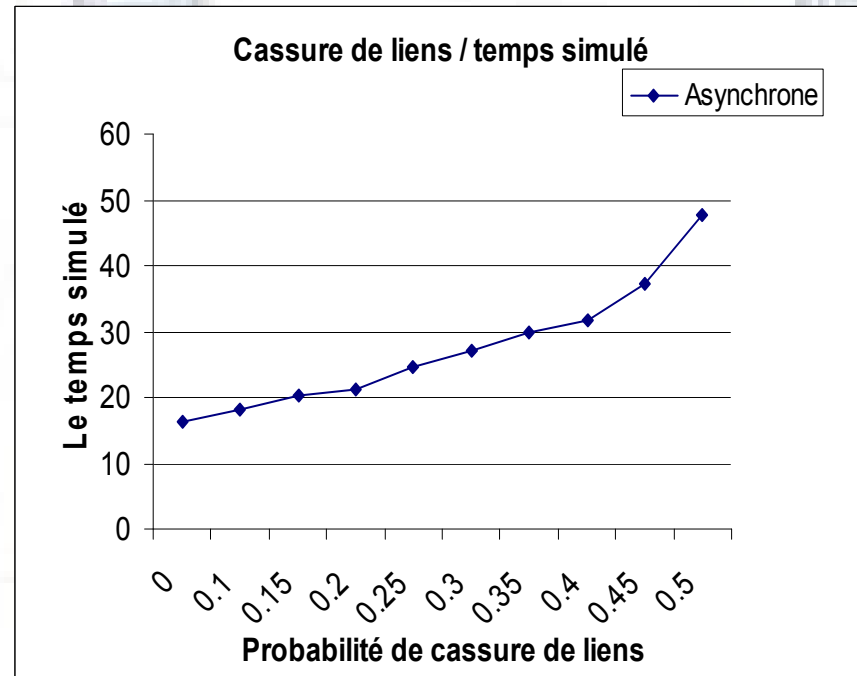
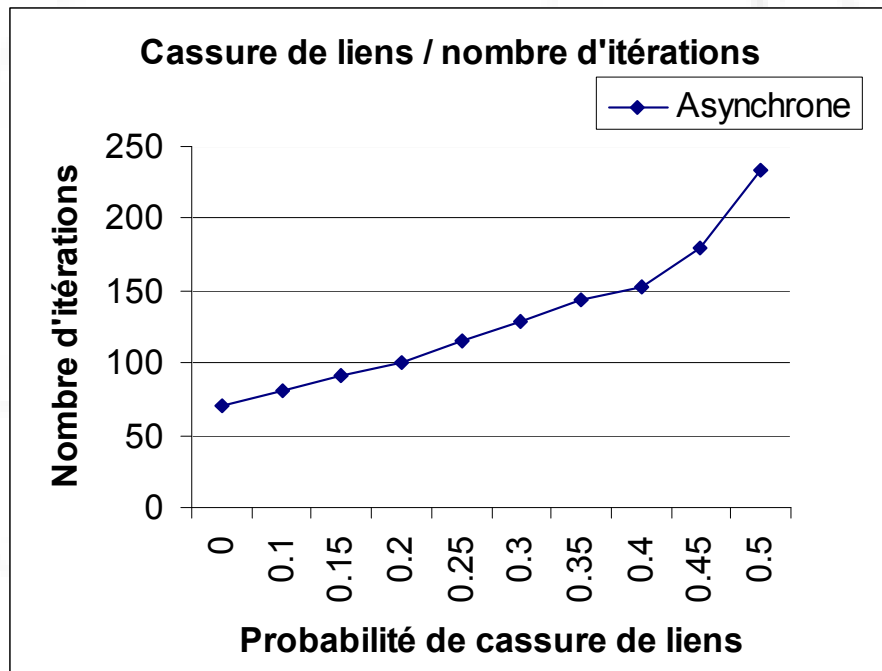
Évaluation

- Passage à l'échelle



Évaluation

- Topologie dynamique



Conclusion

- Les réseaux de capteurs : une technologie récente
 - Contraintes : consommation d'énergie, topologie dynamique, etc.
 - Très orientés vers un domaine d'application précis
 - Problématiques traitées : localisation, couverture, fusion de données
- Localisation :
 - Approche basée sur ancre mobile
 - Choix de la trajectoire de Hilbert
 - Deux algorithmes extensibles et à faible complexité
 - **Limitations : ancre mobile et communications**