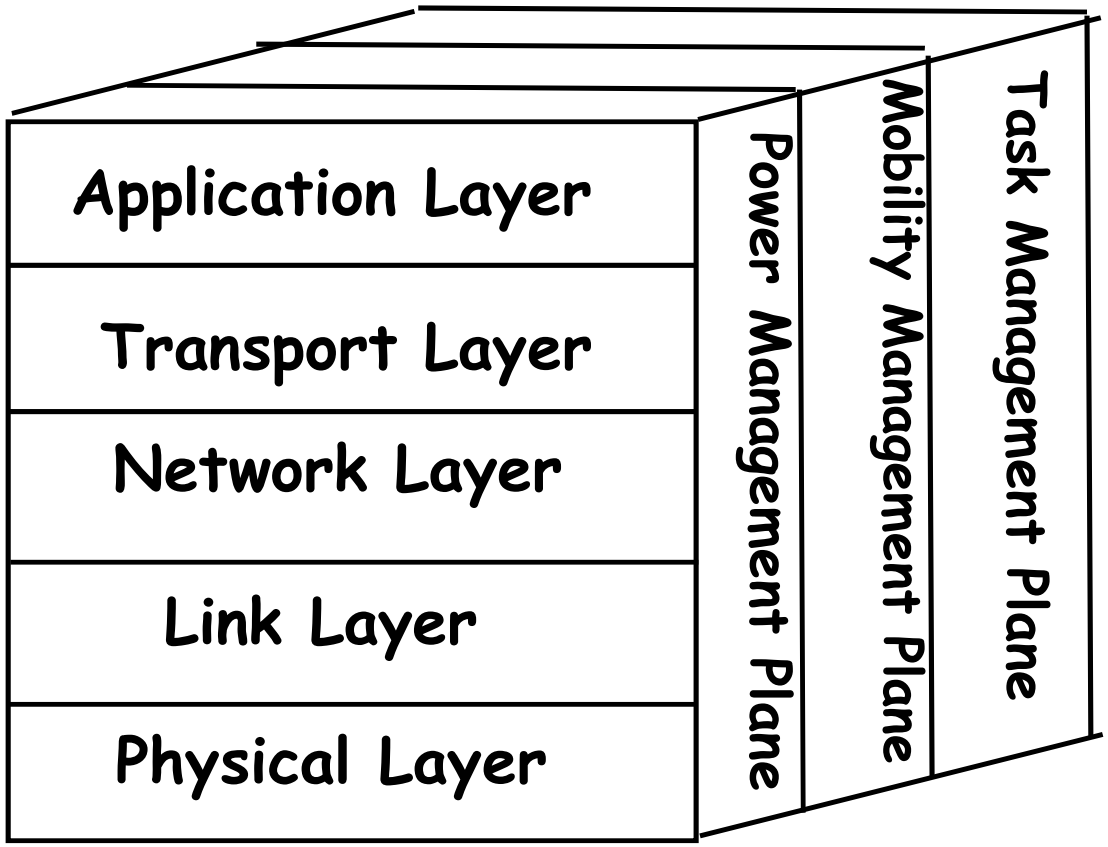
Réseaux de capteurs sans fil

(Choix conceptuels pour le déploiement de l'IoT)

Riadh DHAOU

Riadh.Dhaou@enseeiht.fr

PILE PROTOCOLAIRE



POURQUOI LES PROTOCOLES DES RESEAUX AD-HOC NE PEUVENT PAS ETRE UTILISES POUR LES WSN?

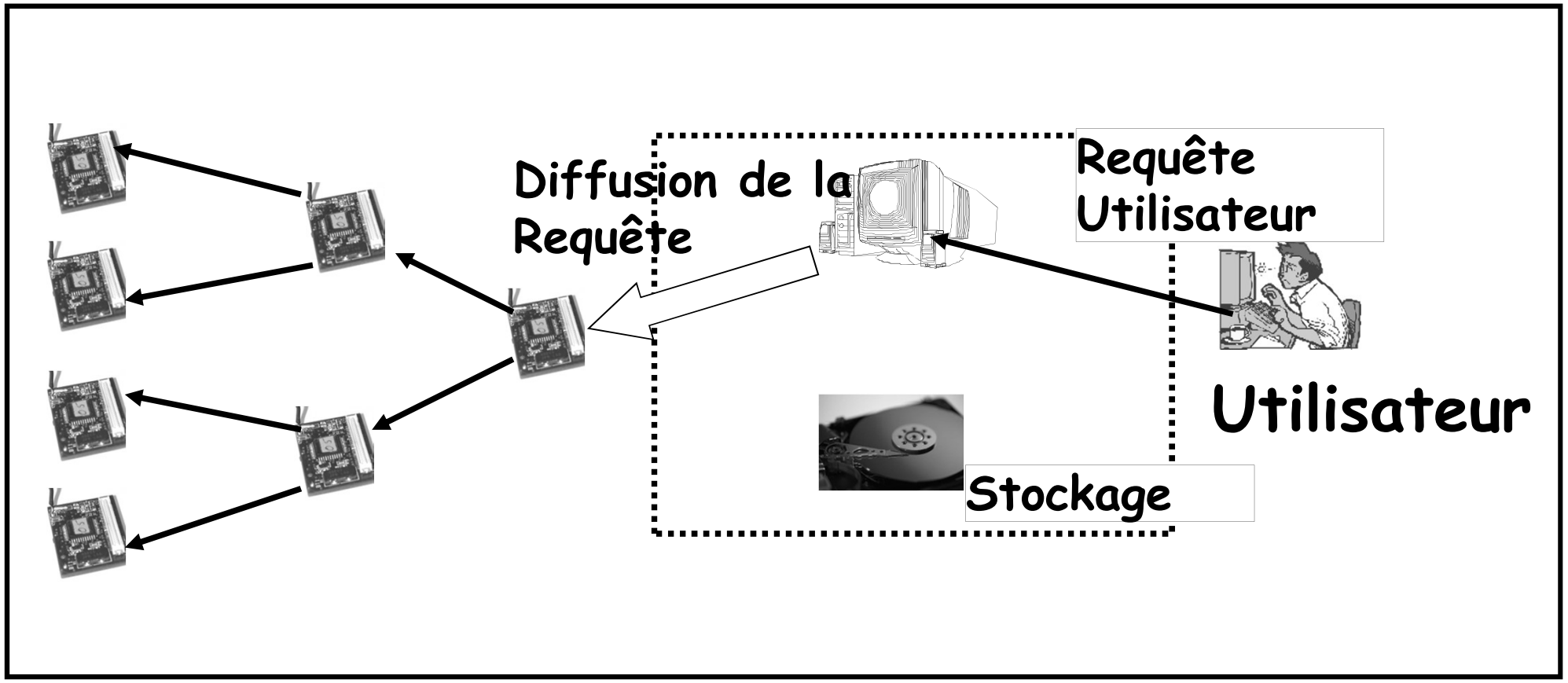
- Le nombre de capteurs est d'un ordre de grandeur plus élevé
- Les capteurs *déployés en masse* et peuvent *tomber en panne*
- La topologie d'un WSN peut changer fréquemment
  - en raison de la mobilité de nœuds et de pannes
- Les capteurs ont une puissance, des capacités de calcul, et une mémoire limitée
- La pile protocolaire TCP/IP complète est trop lourde
- Elle est très dépendante des applications visées

NIVEAU APPLICATION

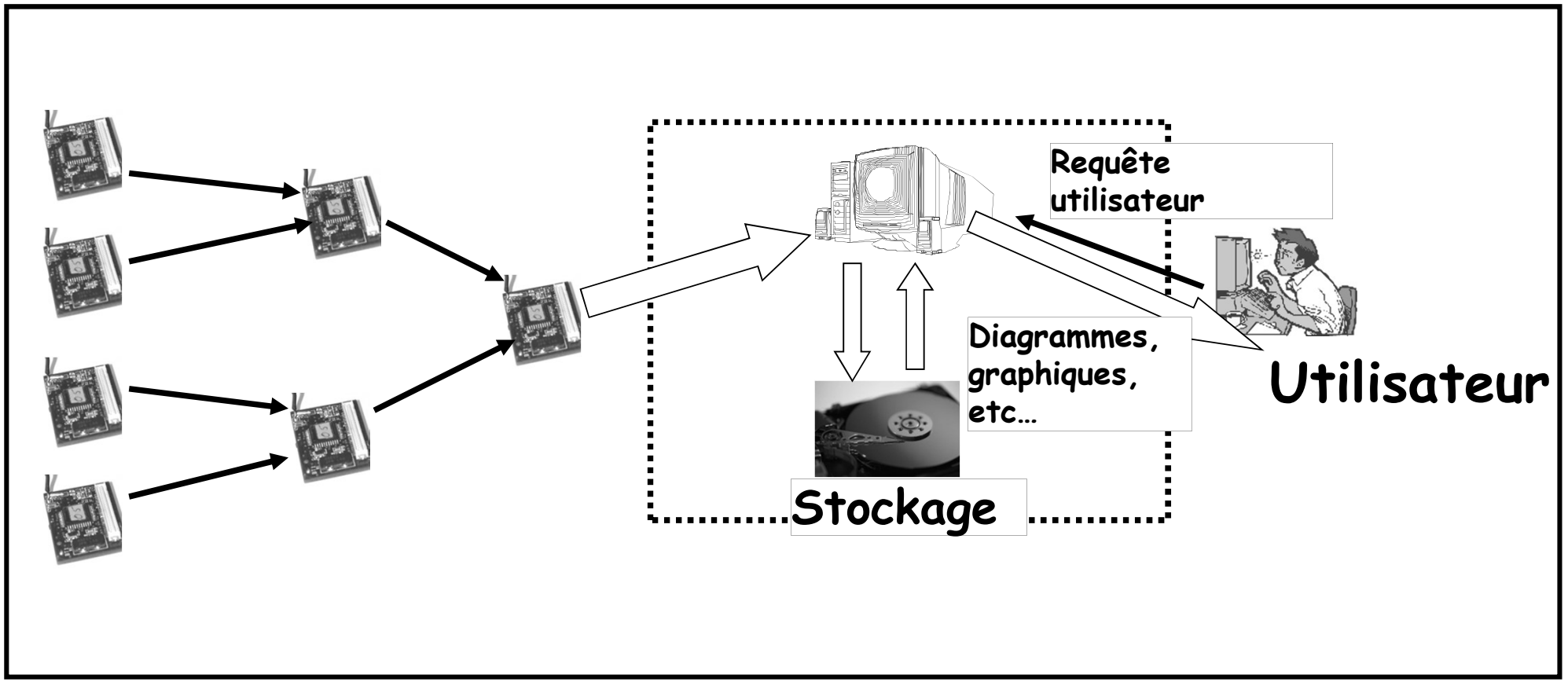
NIVEAU APPLICATON

- Traitement des Requêtes
- Gestion du Réseau de Capteurs

Traitement de Requête



Traitement de Requête



Traitement de Requête

- L'utilisateur envoie une requête et obtient une réponse des capteurs valides, e.g.,

  QUELLE PARTIE A UNE HUMIDITE PLUS QUE 50?

  Type= humidity
  Timestamp=01/18/2007/16:35:28
  Location=[60N,120W]
  Humidity>50

Traitement de Requête

Exemple: (Requête de Poursuite d'Animaux)

Type = four legged animal (detect animal location)
Interval = 30 s (send back events every 30 s)
Duration = 1h (.. for the next hour)
Rec = [-100,100,200,400] (from sensors within the rectangle)

Traitement de Requête

Le capteur détectant l'animal *génère* la donnée suivante:

Type **– four legged animal (type of animal seen)**
Instance**= elephant (instance of this type)**
Location **= (125,220) (node location)**
Intensity **= 0.6 (signal amplitude measure)**
Confidence **= 0.85 (confidence in the match)**
Timestamp**= 01:20:40 (event generation time)**

Un Autre Exemple de Requête

Select [task, time, location, [distinct | all], amplitude,
[[avg | min |max | count | sum] (amplitude)]]
from [any , every]
where [power available [<|>] *PA* |
location [in | not in] *RECT* |
t_{min} < time < t_{max} |
task = *t* |
amplitude [<|==|>] *a*]
group by task
based on [time limit = l_t | packet limit = l_p |
resolution = *r* | region = *xy*]

Sensor Query and Tasking Language (SQTL)

(C-C Shen, et.al., "Sensor Information Networking Architecture and Applications", IEEE Personal Communications Magazine, pp. 52-59, August 2001.)

- SQTL est un langage de scripts procédural.
- Il fournit des interfaces

  pour accéder au hardware (unité de capture):

  - getTemperature, turnOn

  pour localiser le capteur:

  - isNeighbor, getPosition

  et pour communiquer:

  - tell, execute.

Sensor Query and Tasking Language (SQTL)

En utilisant les commandes précédentes, un programmeur peut créer un "bloc de traitement d'évènement" pour 3 types d'évènements:

1. **Événements *générés* quand un message est reçu par un capteur (RECEIVE),**
2. **Événements déclenchés périodiquement (EVERY),**
3. **Événements causés par l'expiration d'un temporisateur (EXPIRE).**

Principales Approches de Traitement de Requête

- Push-based:
  - Le capteur initie la dissémination des informations
- Pull-based:
  - Le Puit initie la dissémination des requêtes
- Push-pull:
  - Les capteurs/puits sont activement impliquées

Principales Approches de Traitement de Requête

- PULL BASED – (INTEREST DISSEMINATION)

  * Les utilisateurs expriment leurs intérêts à un capteur, un sous ensemble de capteurs ou au réseau entier.

  * Cet intérêt peut être au sujet d'un certain attribut du capteur ou d'un évènement de déclenchement

- PUSH BASED – (ADVERTISEMENT OF AVAILABLE DATA)

  * Les capteurs annoncent les données disponibles aux utilisateurs et ces derniers sélectionnent celles qui les intéressent

Classification des Requêtes

- **Requêtes Continues:**

  Collecter des données sur une longue période de temps

- **Requêtes Instantanées:**

  Collecter des données instantanément (ou pour un instant précis)

- **Requêtes Historique:**

  Collecter des données récapitulatives du passé

Types de Requêtes

- **Requête Orientée Data:** (ATTRIBUTE BASED QUERY)

  Recherche de "détection de feu", au lieu d'interrogation individuelle d'un capteur

- **Requête Géographique:** (LOCATION BASED QUERY)

  Reporter les valeurs aux coordonnées {x,y,z}

- **Détection & Control en Temps Réel:**

  Détection d'Intrus

Exemples

- **Data-Centric Query:** (ATTRIBUTE BASED QUERY)

  "Localisation des noeuds qui capturent une température plus élevée que 70° C"

- **Geographical Query:** (LOCATION BASED QUERY)

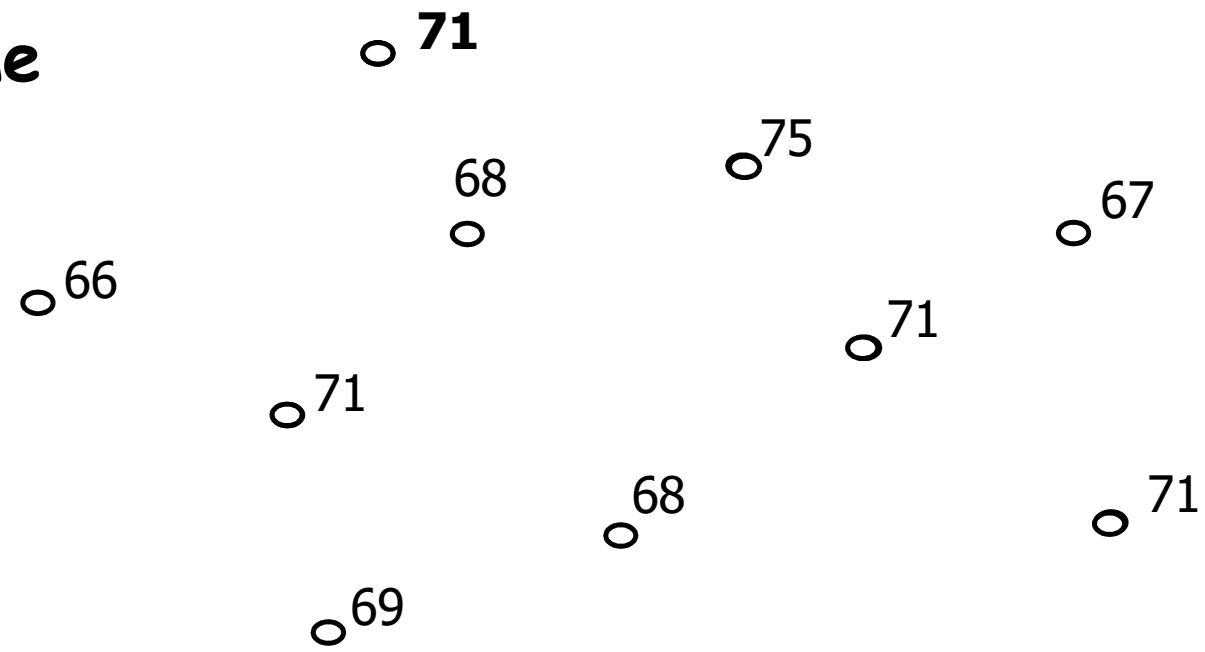
  "Températures relevées par les noeuds de la région A"

Data-Centric Query: (ATTRIBUTE BASED QUERY)

Requête:

Capteur qui relève une température >70°C

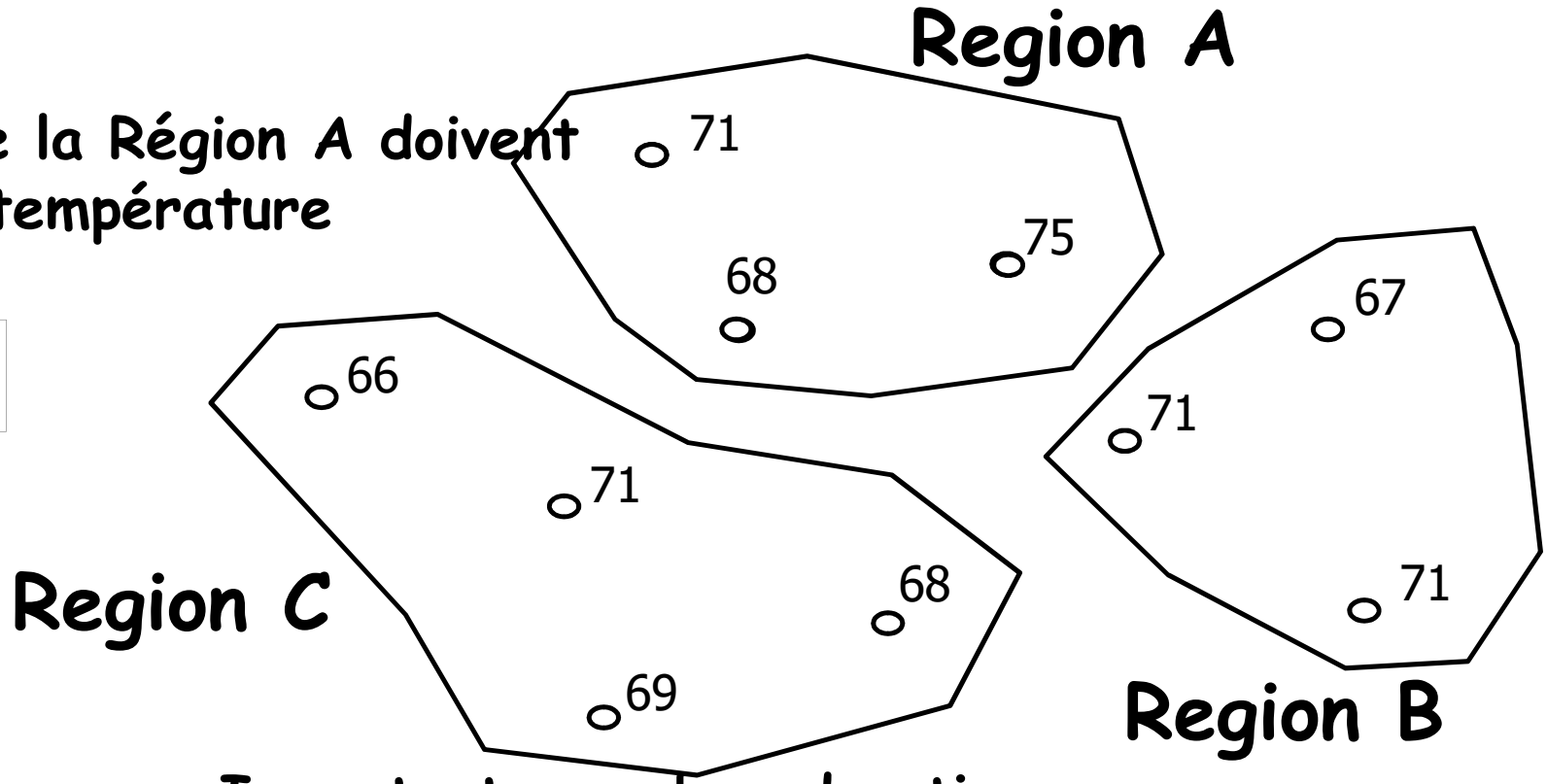
Sink



Geographical Query: (LOCATION BASED QUERY)

Requête:

Les noeuds de la Région A doivent envoyer leur température



Sink

Important pour broadcasting, multicasting, geocasting et anycasting

Agrégation/Fusion de données

- **Impossible de récolter sans interruption les données brutes des capteurs**
  - Mémoires et débits limités
  - Overhead d'Information
- **Les différentes lectures sont d'une utilité limitée:**
  - Intérêt pour une information collective plutôt que différentes données de capteurs
  - Exploiter les traitements en-réseau (network coding?)
- **But:**

  Économiser l'énergie et augmenter la durée de vie du réseau en combinant plusieurs données de capteurs

Agrégation/Fusion de données

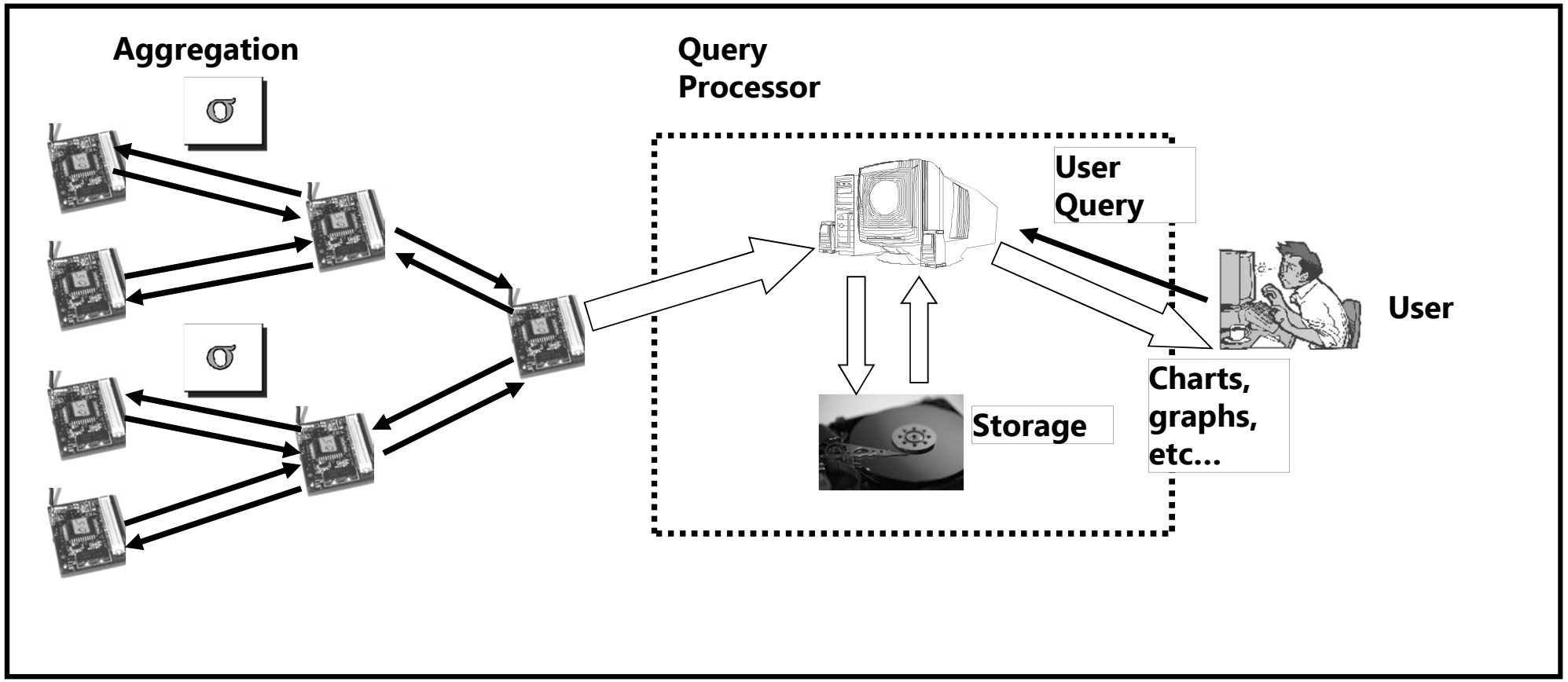
- **Agrégation de Données:**

  Processus de combinaison des données ou informations pour estimer ou prévoir des événements

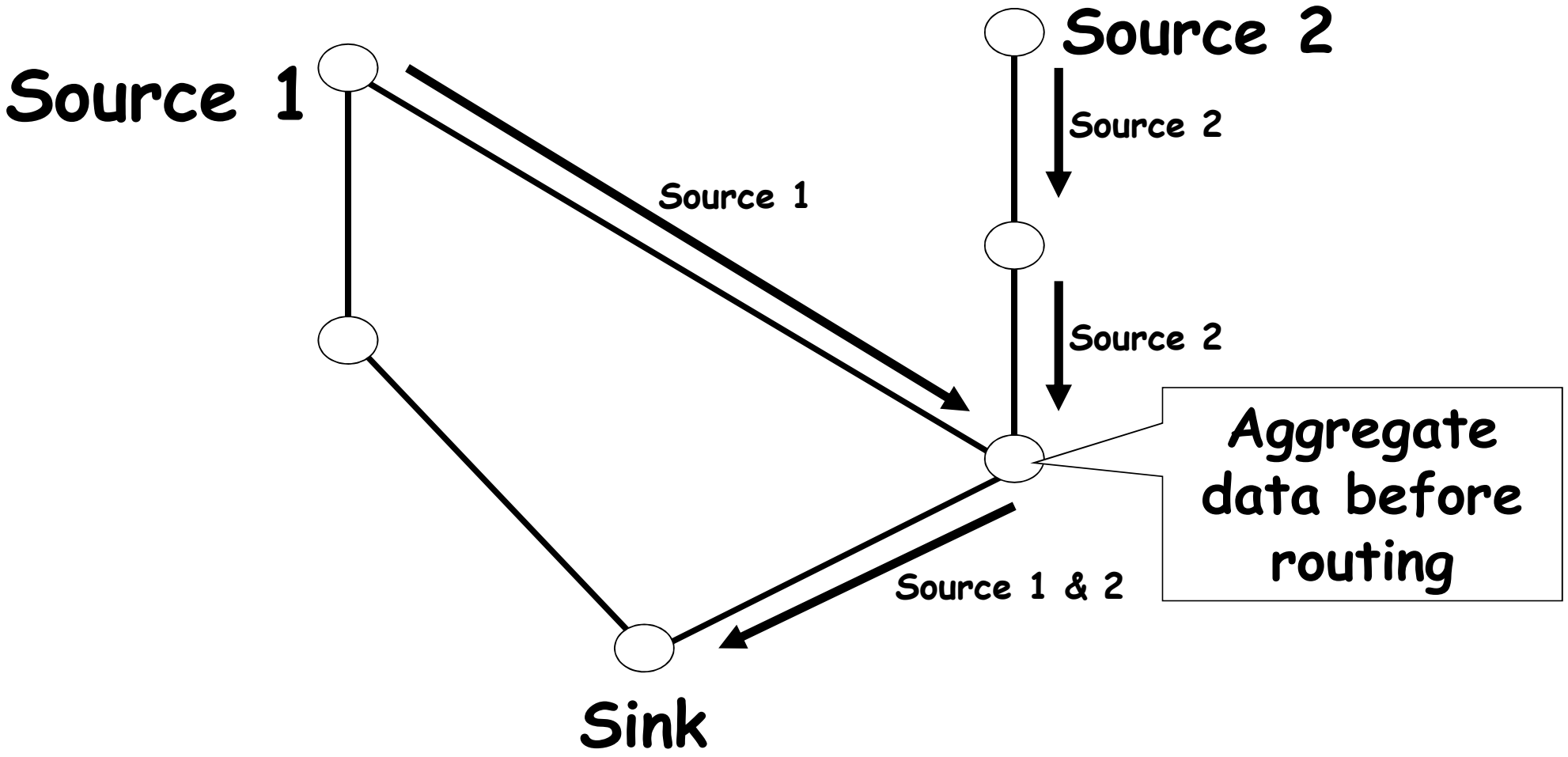
- **Idée:**

  Tirer profit de la hiérarchie de routage et de la densité élevée de réseau

Agrégation/Fusion de données



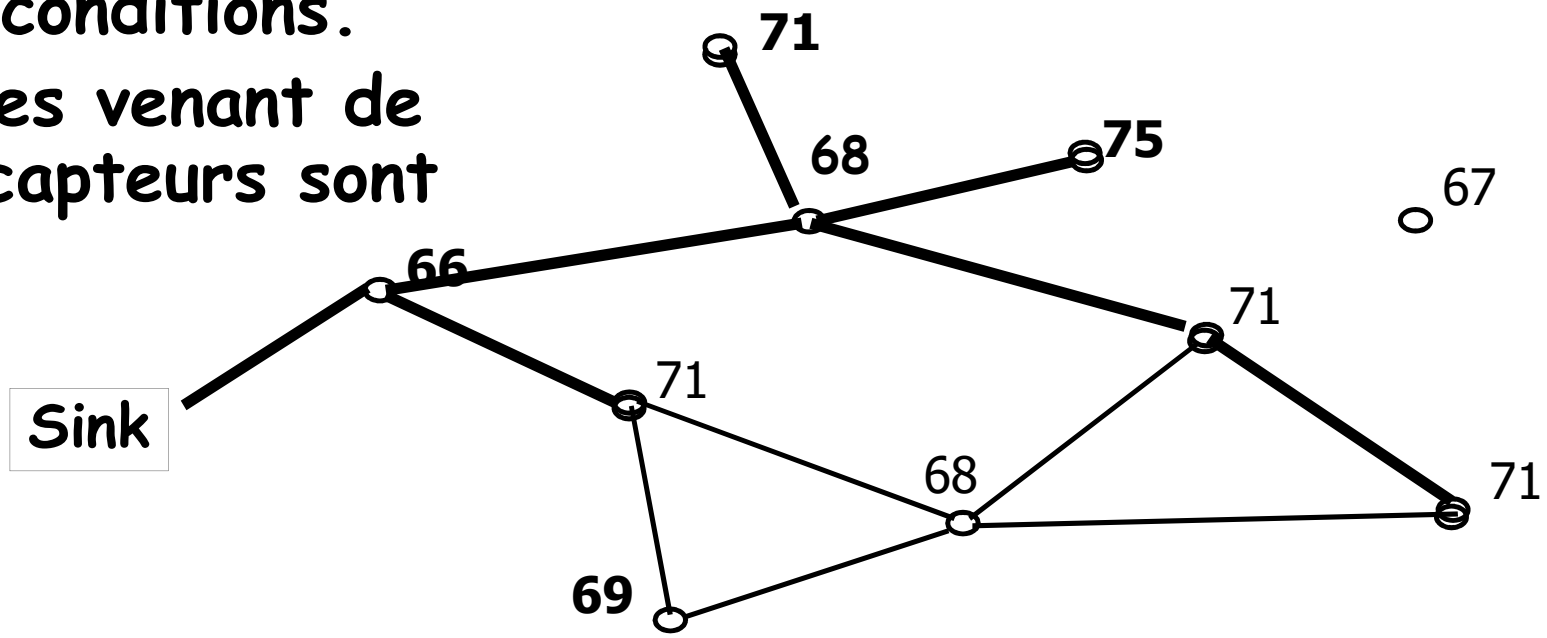
Agrégation de données



Agrégation/Fusion de données

Le puit demande aux capteurs de rapporter sous certaines conditions.

Les données venant de plusieurs capteurs sont agrégées.



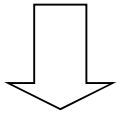
Requête:

Capteurs relevant une température >70°C

Composants d'Agrégation de Données

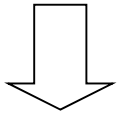
DATA STORAGE

Stocker les données capturées en mémoire de façon efficace, en préservant la précision de l' information.



AGGREGATION FUNCTIONS

Placer des points d'agrégation sur les routes entre les capteurs et le puit.



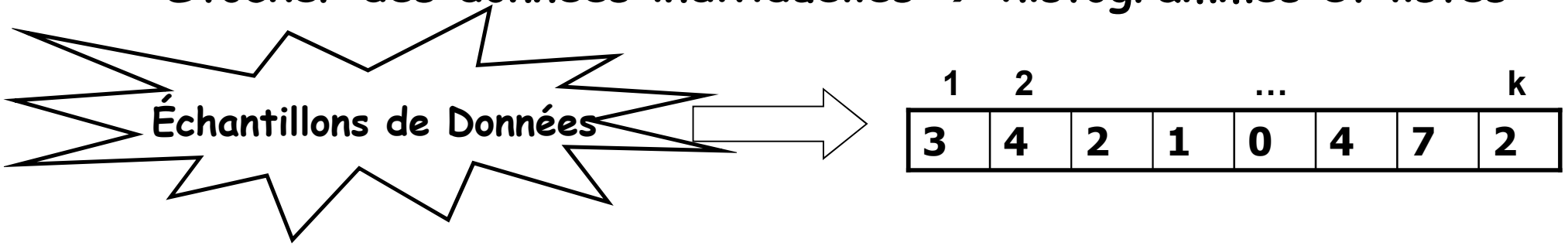
AGGREGATION PATHS

Quels sont les points d'agrégation optimaux? Quel est la route la plus approprié de la source au puit **(favorisant l'agrégation des données)**?

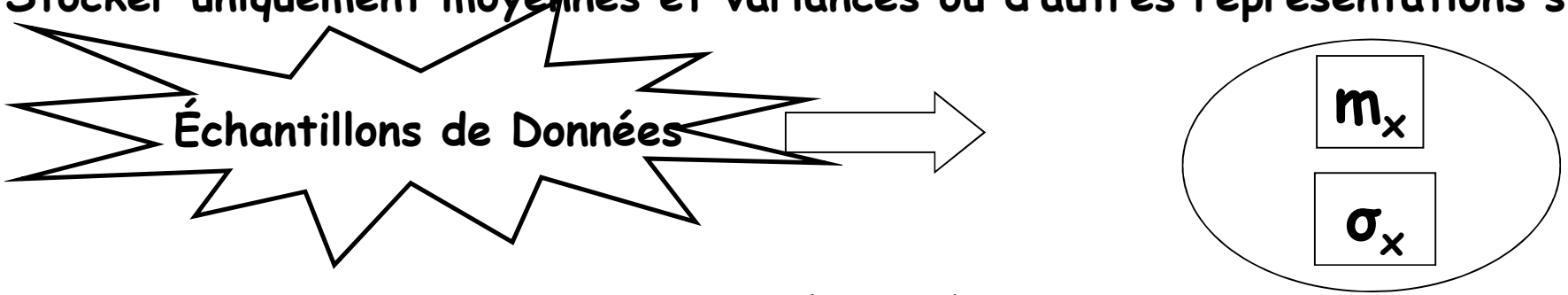
Représentations du Stockage de Données

- Les données peuvent être représentées avec différents degrés de précision.

Stocker des données individuelles → histogrammes et listes



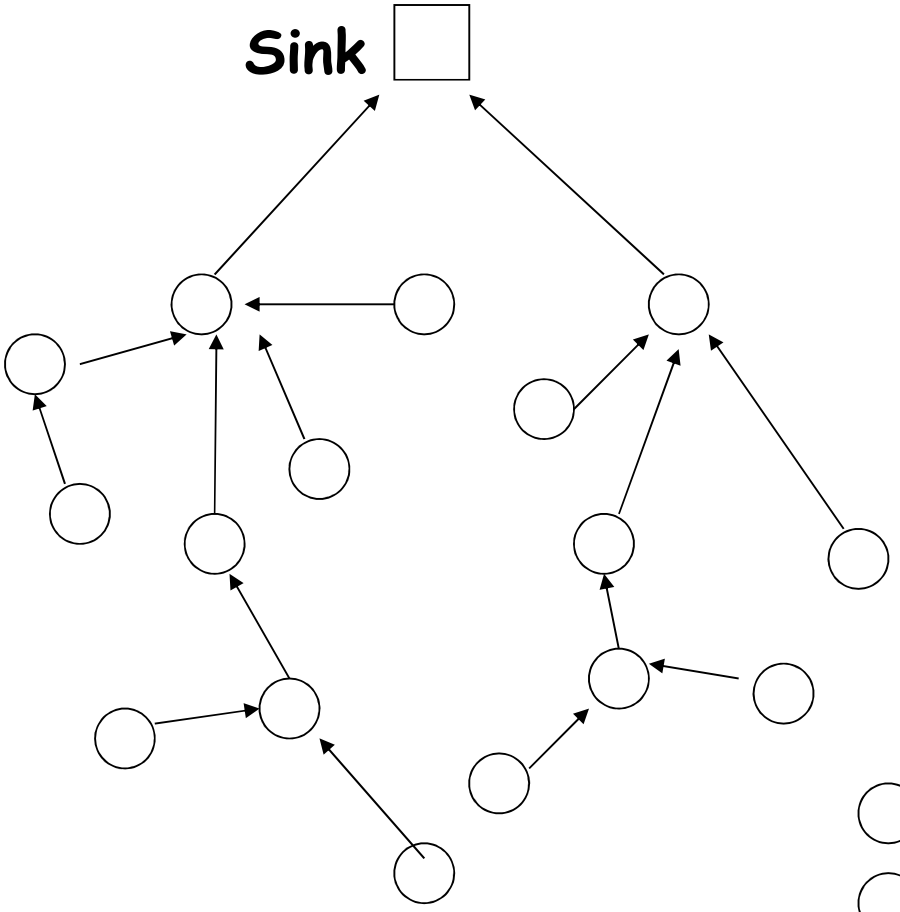
Stocker uniquement moyennes et variances ou d'autres représentations statistiques



Fonctions d'agrégation

- Fonctions très simples
  - Moyenne, max, min, médiane
  - Suppression des doublons
- Fonctions plus sophistiquées
  - Exploiter les corrélations spatiale et temporelle
  - Traitement de signal (convolution, filtrage, etc.)

Chemins d'agrégation



- Défis:
  - Trouver le nombre optimal de points d'agrégation
  - Sélection des points d'agrégation
  - Changement Dynamique des points d'agrégation (efficacité en terme de consommation)

Aggregation point

Sensor node

Difficultés de l'agrégation

- Inévitablement en environnement peu fiable, certaines informations ne sont pas disponibles ou sont chères à obtenir
  - Combien de nœuds présents?
  - Combien de nœuds sont supposés répondre?
  - Quelle est la distribution de l'erreur (en particulier, y a t-il des nœuds malicieux?)
  - Essayer de construire une infrastructure afin d'enlever toute incertitude de l'application peut ne pas être faisable – Voulons-nous établir des transactions distribuées?

Difficultés de l'agrégation

- L'Information s'écoule lentement en un message à la fois
  - Les informations sur le voisinage ne sont jamais complètes et à jour
- Quel type d'information si nous prévoyons de l'agrégation
  - Flux de données (streams)
  - Estimations robustes

Agrégation Optimale de données

- **Agrégation optimale de données est NP-Dur**
- **Algorithmes Sous-Optimaux:**
  - **Opportunistes**
    - **Agréger juste quand c'est possible**
  - **Center at Nearest Source (CNS)**
    - **La source la plus proche agit en tant que point d'agrégation**
  - **Shortest Paths Tree (SPT)**
    - **Les sources envoient en utilisant le chemin le plus court** (s'il permet d'agréger)
  - **Greedy Incremental Tree (GIT)**
    - **Sélectionner Récursivement la source la plus proche a l'arbre**
  - **Clustered Diffusion with Dynamic Data Aggregation**
    - **Hybride entre diffusion et agglomération (clustering) avec une capacité d'agréger les données dans les "cluster heads"**

Avantages et Inconvénients de l' Agrégation

- **Avantages:**
  - L'agrégation en cours de route des données réduite la charge de trafic
    - Efficacité en terme de consommation électrique et de mémoire
  - Passage à l'échelle (aussi bien pour un grand nombre de puits que de capteurs)

- **Inconvénients:**
  - Nécessite une conception prenant en compte un compromis entre précision, stockage et taille de message
  - Le risque de perte d'information, rend l'estimation robuste plus difficile:
    - Ex. une perte d'une simple donnée lue peut endommager les agrégats MAX/MIN

Difficultés d'interrogation

- **Haute densité:** Données Riches et Massives
- **Corrélation:** Données réparties et corrélées spatialement
- **Incertitude:** Bruit, données erronées, des témoins.
- **Sémantique:** Données individuelles ≠ information utile

Interroger les Réseaux de Capteurs

- TinyDB
  - Cadre pour le stockage d'information
  - Collection de données fondée sur un arbre
- *COUGAR*
  - Données de capture individuelles
  - Rassemblement Distribué
  - SQL-like
- TAG (Tiny Aggregation)
  - Focalise sur L'Agrégation en utilisant un langage à requête SQL-like
  - Intégré à TinyOS

TinyDB

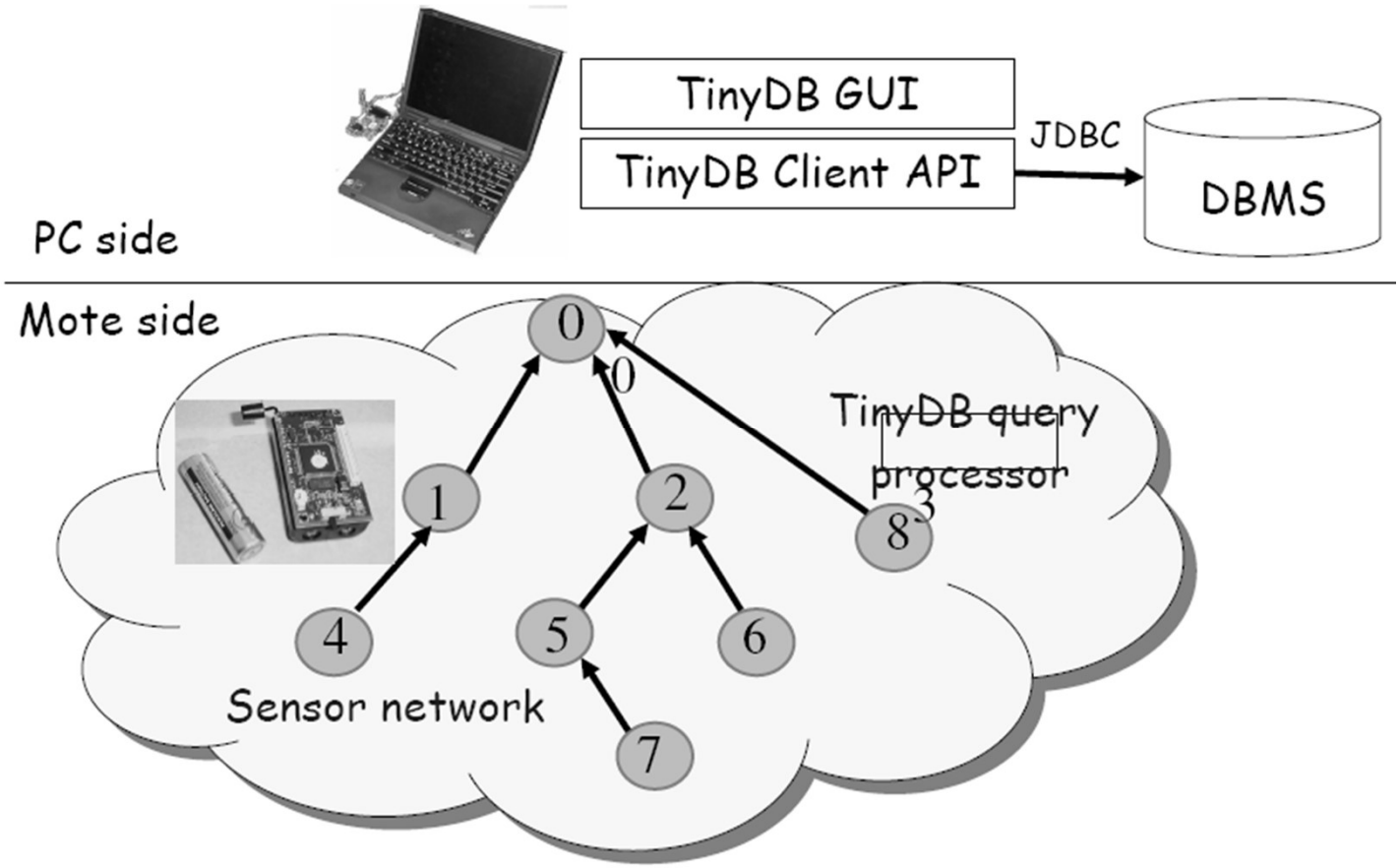
S. R. Madden, et. al., ``TinyDB: An Acquisitional Query Processing System for Sensor Networks,'' *ACM Transactions on Database Systems*, pp. 122–173, March 2005.

- Un système de gestion de requête distribué pour un réseau de Mica motes (UC Berkeley)
- Objectif: Élimine le besoin d'écrire du code C pour la plupart des utilisateurs de TinyOS
- Caractéristiques
  - Requêtes Déclaratives
  - Opérations temporelles + spatiales
  - Routage Multihop
  - Stockage dans le réseau

TinyDB

- Réseau de Capteurs = Base de données distribuée
- Données stockées localement
- Structure du Réseau: routage fondé sur arbre
- Top-down SQL query
- Résultats agrégés retournés au puit

Architecture TinyDB

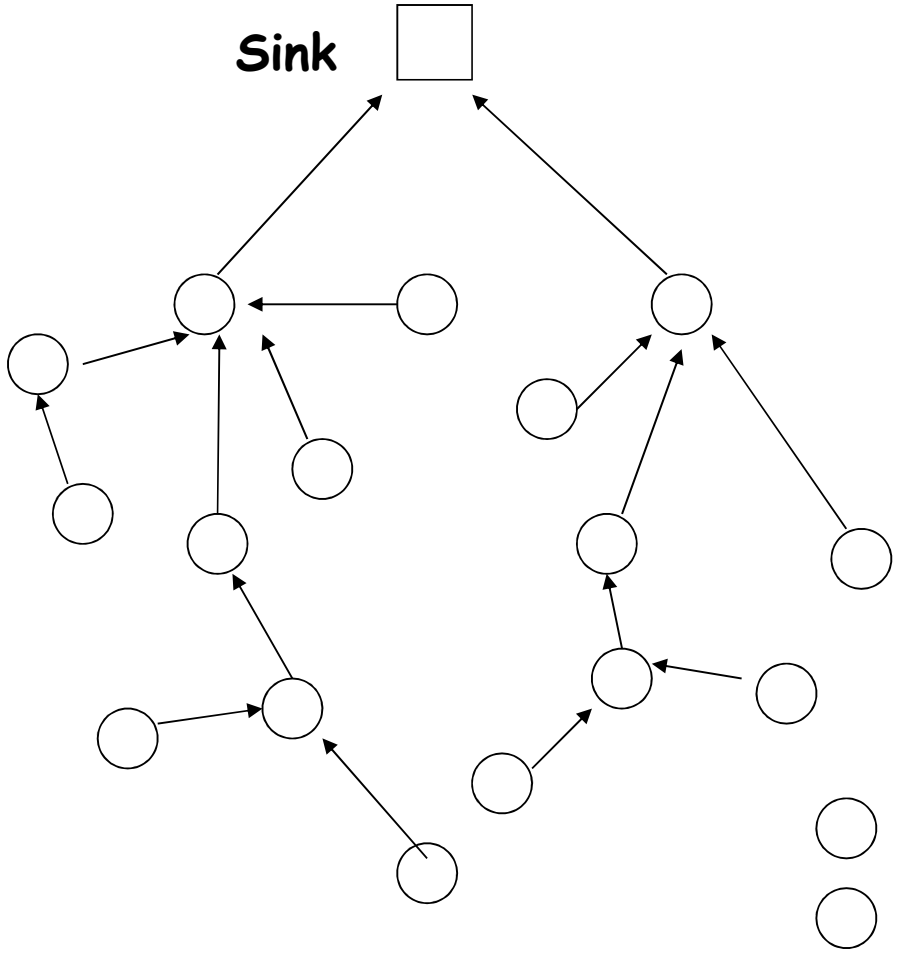


COUGAR

S. Madden and J. Gehrke, ``Query Processing in Sensor Networks,'' *IEEE Pervasive Computing*, vol. 3, No. 1., March 2004.

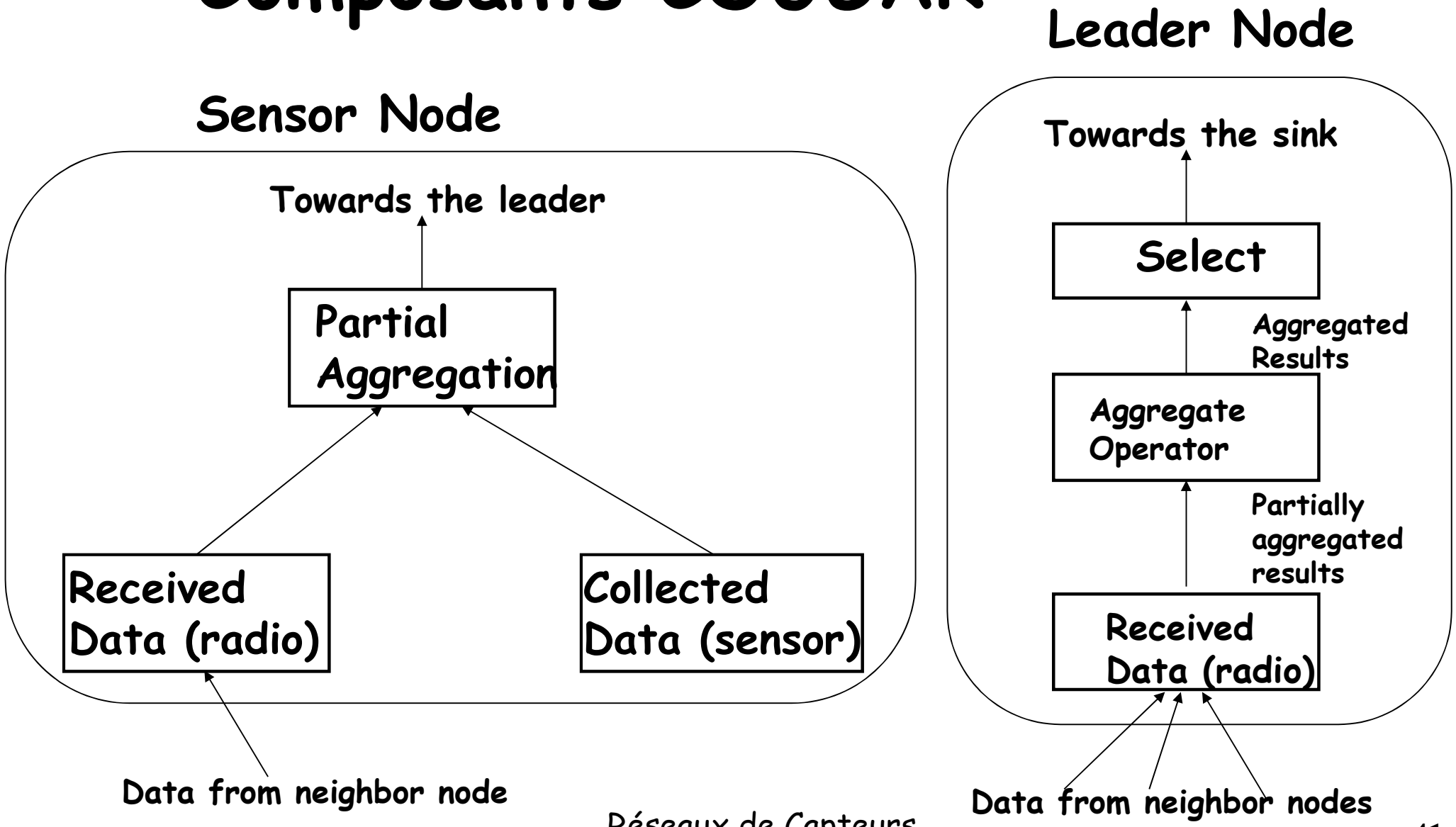
- Exécute des calculs dans le réseau
- Réduire la consommation électrique
- Essaye de fusionner les requêtes similaires
- Propage les résultats au puit

Architecture COUGAR



- Les capteurs envoient leurs informations vers les leaders désignés
- Agrégation partielle au niveau des capteurs
- Agrégation optimale des données collectées au niveau des leaders

Composants COUGAR



Selection de leaders COUGAR

- Méthodes
  - Fixes
  - Aléatoires
- Politique de sélection des Leaders
  - Maintenance Dynamique en cas de panne
  - Minimiser la distance de communication

Tiny Aggregation (TAG)

S. Madden et al., "TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks" , in Proc. of OSDI, 2002

- Service d'agrégation pour TinyOS capteurs "motes"
- Langage déclaratif SQL-like
- Utilise des interfaces simples et déclaratives
- Sensible aux contraintes sur les ressources et pannes des liens
- Réduit les coûts en fonction des types d'agrégats

Approche TAG

- Distribution des Requêtes: Les requêtes d'agrégations sont diffusées (push-down) dans le réseau afin de construire un spanning tree à partir du puit
  - La racine diffuse la requête, chaque noeud écoutant le requête la diffuse.
  - Chaque noeud sélectionne un parent. La structure de routage est un spanning tree avec comme racine le noeud source de la requête.
- Collection de Données: Les valeurs agrégées sont routées sur l'arbre.
  - Les nœuds internes agrégent les données partielles reçues de leurs sous-arbres.

NIVEAU RESEAU
(ALGORITHMES DE ROUTAGE)

Pourquoi ne pouvons-nous pas employer des algorithmes conventionnels de routage ici??

- **Un capteur n'a pas d'identité (adresse)**
  - Routage fondé sur le Contenu **et** Orienté données
    - *** Où sont les noeuds avec une température dépassant les 10 degrés pour les prochaines 10 minutes?**
    - *** Donner la localisation d'un objet (avec l'expression d'un intérêt) chaque 100ms pendant 2 minutes.**

Pourquoi ne pouvons-nous pas employer des algorithmes conventionnels de routage ici??

- Plusieurs capteurs collaborent pour atteindre un but.
- Les noeuds intermédiaires peuvent en plus du routage faire une agrégation et/ou sauvegarde en cache de données.

  * Où, Quand, Comment?

Pourquoi ne pouvons-nous pas employer des algorithmes conventionnels de routage ici??

- Pas de commutation de paquet de proche-en-proche, mais plutôt une propagation de données de noeud-en-noeud.

- Tâches de haut niveau sont requises:
  * Vers quelle direction et à quelle vitesse se déplace l'éléphant?
  * Est-ce l'heure de commander plus d'inventaires?

Contraintes

- Noeuds avec une source d'énergie limitée
- Calculs
  - Agrégation de données
  - Supprimer les informations de routage redondantes
- Communication
  - Débit limité
  - Grande consommatrice d'énergie

But: Réduire la dissipation d'énergie

Contraintes

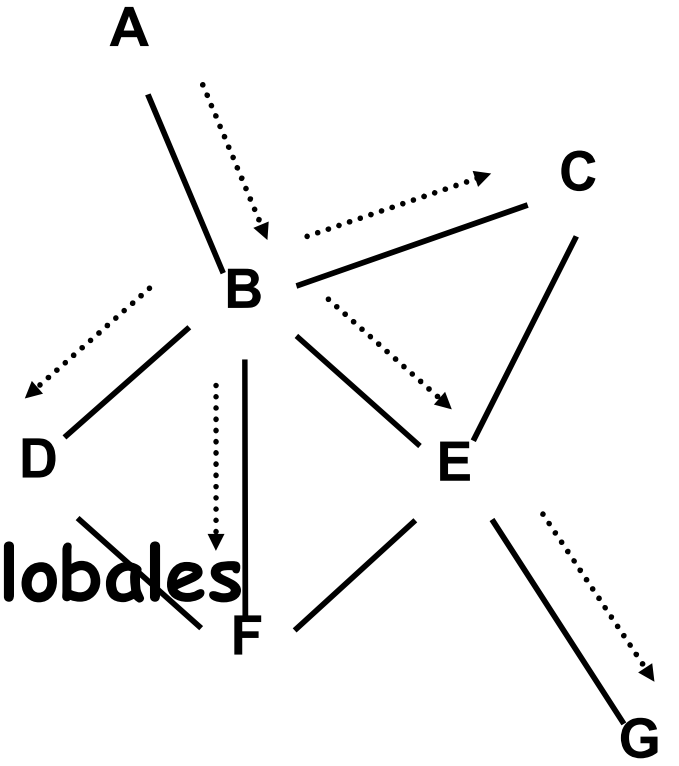
- Passage à l'échelle: déploiement ad-hoc à large échelle
  - Entièrement distribué sans connaissance globale
  - Grand nombre de sources et de puits
- Robustesse: pannes inattendues de noeuds capteurs
- Changements Dynamiques: aucune connaissance *à-priori*
  - Mobilité du puit
  - Mobilité de la cible

Contraintes

- Problèmes Topologiques ou *géographiques*
- Le temps: les données périmées n'ont pas de valeur
- La valeur des données est fonction du temps et de la position
- Y-a-t il un besoin de techniques *générales* pour différentes applications de capteurs?
  - Capteurs utilisant de petites puces
  - Capteurs étendus *géographiquement*
  - Capteurs mobiles, e.x., robots

Quelles propriétés doit vérifier le Protocole de Routage Optimal "Idéal" pour les WSNs

- Routes avec le plus court chemin
- Éviter le chevauchement
- Consommer le minimum d'énergie
- Besoin d'informations topologiques globales



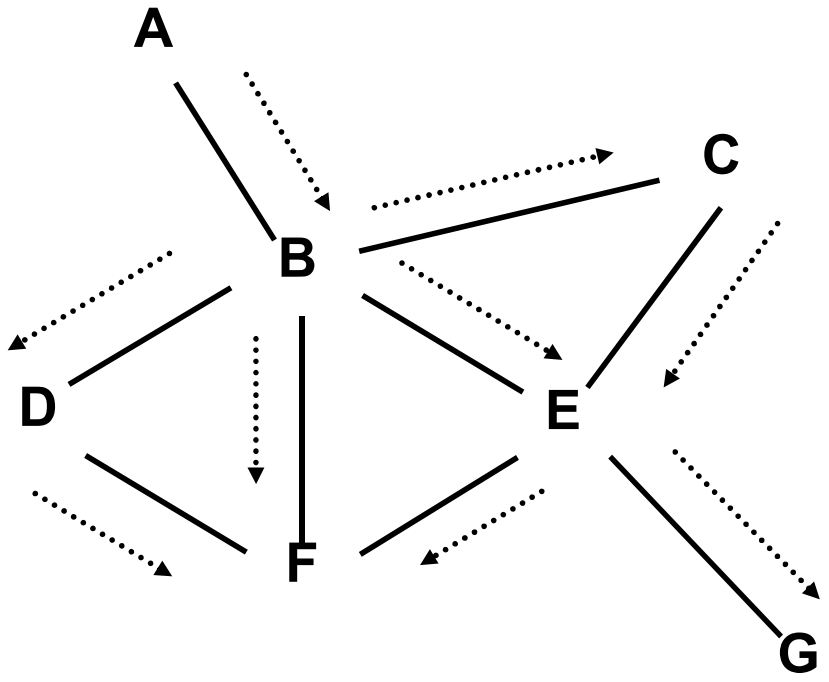
Classification des Protocoles de Routage pour les WSN

K. Akkaya and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks," AdHoc Networks (Elsevier) Journal, 2005

1. DATA CENTRIC PROTOCOLS: Flooding, Gossiping, SPIN, Directed Diffusion, SAR (Sequential Assignment Routing) , Rumor Routing, Constrained Anisotropic Diffused Routing, COUGAR, ACQUIRE
2. HIERARCHICAL PROTOCOLS: LEACH, PEGASIS, TEEN (Threshold Sensitive Energy Efficient Sensor Network Protocol), APTEEN,
3. LOCATION BASED (GEOGRAPHIC) PROTOCOLS: MECN, SMECN (Small Minimum Energy Com Netw), GAF (Geographic Adaptive Fidelity), GEAR, Distributed Topology/Geographic Routing Algorithm (PRADA)

Approche Conventionnelle: FLOODING

Diffuse les données à tous les noeuds voisins



Gossiping

Envoie les données à un voisin sélectionné aléatoirement.

Problèmes de Flooding and Gossiping

PROBLEMES:

Bien que ces techniques sont simples et réactives, elles ont quelques inconvénients comprenant:

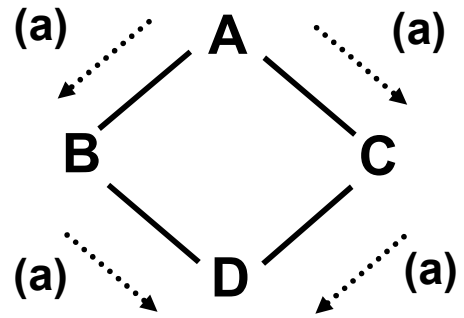
- **Implosion**

  (NOTE: Gossiping évite ceci en choisissant seulement un noeud; mais ceci cause des délais pour propager les données à travers de réseau)

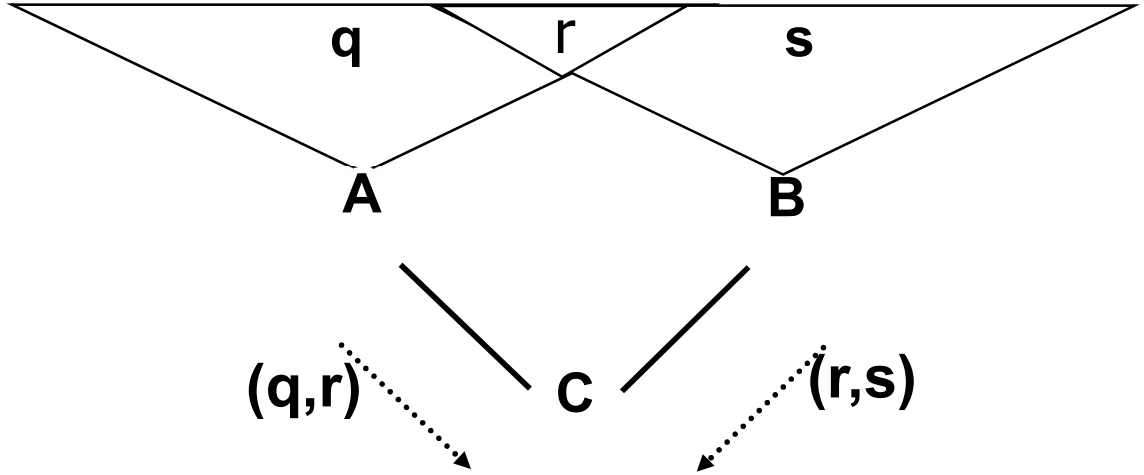
- **Chevauchement**
- **Épuisement de ressources**
- **Consommation (énergie) inefficace**

Problèmes

Implosion



Chevauchement des Données



Épuisement des Ressources

Aucune connaissance de l'énergie disponibles des ressources

Gossiping

- Utilise un choix aléatoire pour économiser l'énergie
  Sélectionner aléatoirement un seul noeud et lui envoyer les données
- Évite les implosions
- Distribue l'information lentement
- Dissipe l'énergie lentement

SPIN: Sensor Protocol for Information via Negotiation

W.R. Heinzelman, J. Kulik, and H. Balakrishnan,"Adaptive Protocols for Information Dissemination in Wireless Sensor Networks", *Proc. ACM MobiCom'99*, pp. 174-185, 1999;

* Utilise trois types de messages: ADV, REQ, et DATA.

* Quand un noeud capteur a quelque chose de nouveau, il diffuse un « packet advertisement » (ADV) qui contient la nouvelle donnée, e.x., des méta données.

* Les noeuds intéressés envoient un « packet request » (REQ).

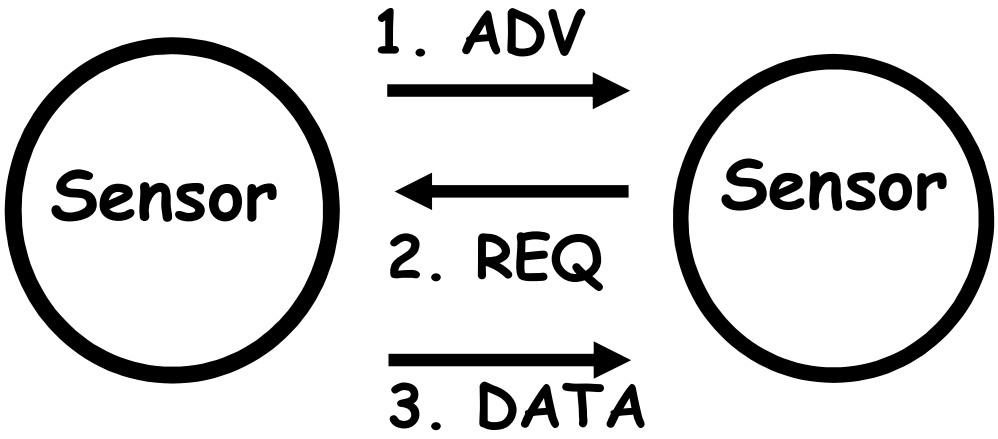
* Les données sont envoyées aux noeuds qui l'ont demandé en utilisant des paquets de données DATA.

* Ceci sera répété jusqu'à ce que tous les noeuds obtiennent une copie.

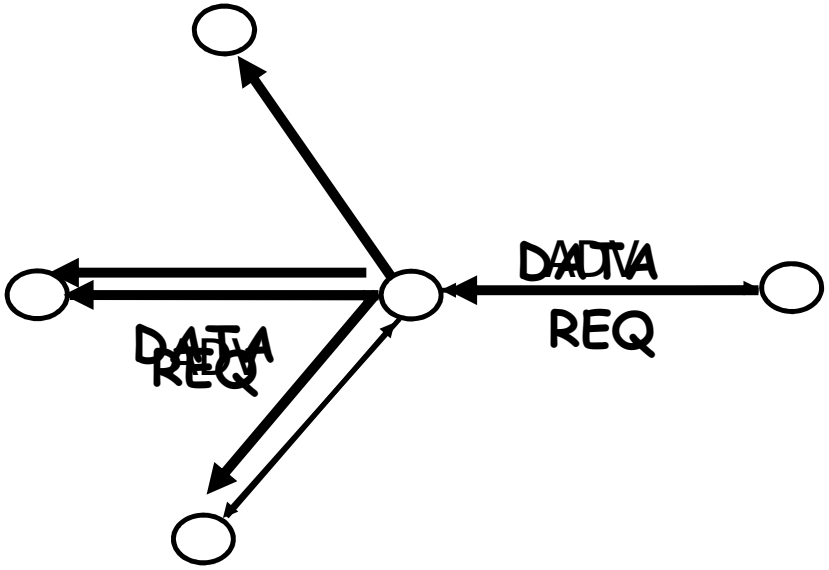
SPIN

- Convient à la diffusion d'information pour tous les noeuds capteurs.

- SPIN est fondé sur un routage orienté données où les capteurs diffusent un « advertisement » pour les données disponibles et attendent une requête des puits intéressés



SPIN



Famille de Protocoles SPIN

- SPIN-PP – Pour communication point à point
- SPIN-EC – Semblable à SPIN-PP mais avec l'ajout d'heuristique de conservation d'énergie
- SPIN-BC – Conçu pour les réseaux à diffusion. Les noeuds arment des timers aléatoires après la réception de ADV et avant l'envoi de REQ pour attendre d'autres noeuds pour envoyer la REQ
- SPIN-RL – Semblable à SPIN-BC avec l'ajout de fiabilité. Chaque noeud maintient s'il reçoit des requêtes de données dans un délai limite, sinon, la donnée est re-demandée

Protocol SPIN-PP

- SPIN-PP
  - Protocole 3-stage handshake
  - Avantages
    - Simple
    - Évite l'Implosion
    - Coût d'Initialisation Minimal
  - Inconvénient
    - * Ne peut pas isoler les noeuds qui ne veulent pas recevoir d'information.
    - * Consomme inutilement de l'énergie.

SPIN-EC

- Spin-EC
  - SPIN-PP + Seuil à énergie réduite
  - Modifie le comportement en fonction des ressources énergétiques courantes

SPIN-EC

- Ajoute une heuristique de conservation d'énergie simple
- Quand l'énergie est abondante, SPIN-EC se comporte comme SPIN-PP
- Quand l'énergie approche un seuil d'énergie réduite, le noeud de SPIN-EC réduit sa participation au protocole (DORMANT)
  - Ne participe à une étape du protocole que si et seulement si le noeud peut « potentiellement » accomplir toutes les étapes restantes

SPIN- CONCLUSIONS

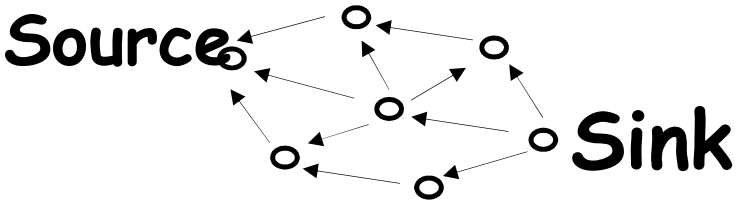
- Flooding converge en premier
  - Sans délais
- SPIN-PP
  - Réduit la consommation d'énergie de 70%
  - Pas de messages DATA redondants
- SPIN-EC distribue
  - 10% plus de données par unité d'énergie que SPIN-PP
  - 60% plus de données par unité d'énergie que flooding

SPIN- CONCLUSIONS

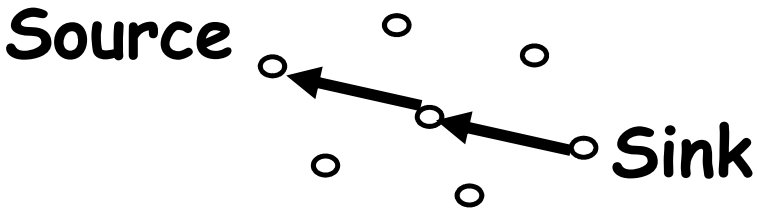
- Energie
  - Plus efficage que flooding
- Latence
  - Converge Rapidement
- Passage à l'échelle
  - Seulement interactions locales
- Robustesse
  - Immunisé contre les pannes de noeuds

Directed Diffusion Routing Algorithm

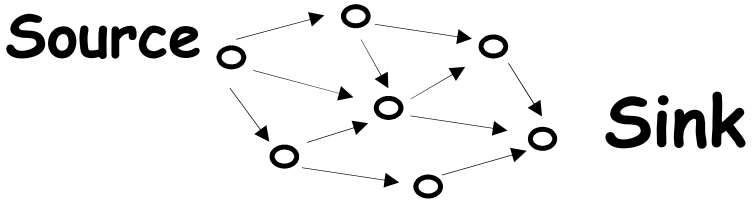
C. Intanagonwiwat, et.al.,
"Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", IEEE/ACM Transactions on Networking, 2003.



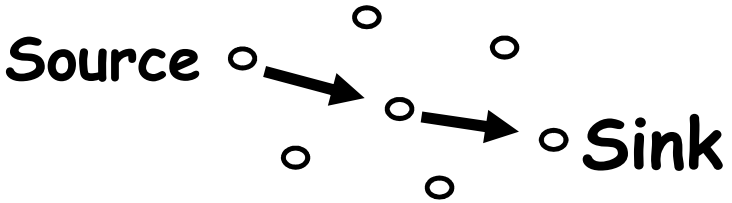
Interest Propagation



Reinforcement



Gradient Setup



Data Delivery

Principes de base

- DATA CENTRIC ROUTING Plan de routage orienté donnée!!!!

- Très grand nombre de capteurs → impossible d'attribuer des IDs spécifiques.

- Sans identifiant unique, collecter les données devient problématique

- Pour résoudre ce problème, certains protocoles de routage rassemblent/routent les données en utilisant la description des données e.x., data-centric.

Qu'est ce qu'un routage orienté donnée (DATA CENTRIC)?

- Data-Centric
  - Pas besoin d'identité du nœud capteur !!!
    - Quelle est la température au noeud #27 ?
  - Les Données sont nommées par des attributs
    - Quels sont les nœuds pour lesquels la température a dépassé récemment les 30 degrés ?
    - Combien de piétons observez vous dans la région X?
    - Dites moi dans quelle direction ce véhicule dans la région Y se déplace t-il?

Qu'est ce qu'un routage orienté donnée (DATA CENTRIC)?

Requière un nommage fondé sur des attributs quand les utilisateurs sont plus intéressés à la valeur d'un attribut pour un phénomène, qu'a interroger individuellement un noeud.

Exemple:

la requête

"Le secteur ou la température dépasse 70° C"

est plus commune que la requête

"La température lue par un certain noeud (e.g., #27)".

Éléments de « Directed Diffusion »

- Le plan de nommage (Naming scheme)
  - *La Donnée est nommée en utilisant un couple* attribue-valeur
- Les requêtes d'Interet (Interests)
  - *Un noeud demande des données en envoyant/exprimant ses « intérêts » pour des données nommées*
- Les Gradients (Gradients)
  - *Les Gradients* sont installés dans le réseau conçu pour "lever" des *événements (e.x. données répondant aux intérêts).*
- Renforcement (Reinforcement)
  - *Le puit renforce des voisins particuliers, pour lever des événements avec une plus haute qualité (débit de données plus élevé)*

NAMING SCHEME

*** La donnée *générée* par un noeud capteur est NOMMEE par un couple ATTRIBUE-VALEUR**

*** Afin de créer une requête, un « intérêt » est défini en utilisant une liste de paires attribue-valeur tels que le nom des *objets, intervalle, durée, zone géographique, etc.***

*** Un noeud capteur quelconque (habituellement le PUIT) utilise le couple attribue-valeur (interests) pour interroger les données des capteurs à la demande**

NAMING SCHEME

Requête: Interest (Task) Description

Exemple: (Animal Tracking Task)

Type = four legged animal (detect animal location)

Interval = 30 s (send back events every 30 s)

Duration = 1h (.. for the next hour)

Rec = [-100,100,200,400] (from sensors within the rectangle)

NAMING SCHEME

Les données envoyées en réponse aux « intérêts » sont nommées pareillement.

Exemple: REPLY

Le capteur détectant l'animal produit les données suivantes:

Type **– four legged animal (type of animal seen)**
Instance**= elephant (instance of this type)**
Location **= (125,220) (node location)**
Intensity **= 0.6 (signal amplitude measure)**
Confidence **= 0.85 (confidence in the match)**
Timestamp**= 01:20:40 (event generation time)**

INTERESTS

*** Le puit diffuse périodiquement un intérêt aux noeuds capteurs pour les interroger sur une information d'une zone particulière du domaine.**

*** Pendant que l'intérêt se propage, la donnée peut être *transformée localement* (e.g., *agrégée*) à chaque noeud, ou peut être *stockée en cache*.**

*** Chaque noeud maintient un cache d'intérêts**

*** Chaque élément correspond à un intérêt distinct**

*** Agrégation d'intérêt: type identique, complètement, rectangle de chevauchement, attribue**

INTERESTS

* Chaque entrée dans le cache a plusieurs champs
  * *Timestamp*: correspondant au dernier intérêt
  * *Plusieurs gradients*: débit de données, durée, direction

* Les autres noeuds peuvent exprimer des « *intérêts* » fondés sur ces attributs

Interests

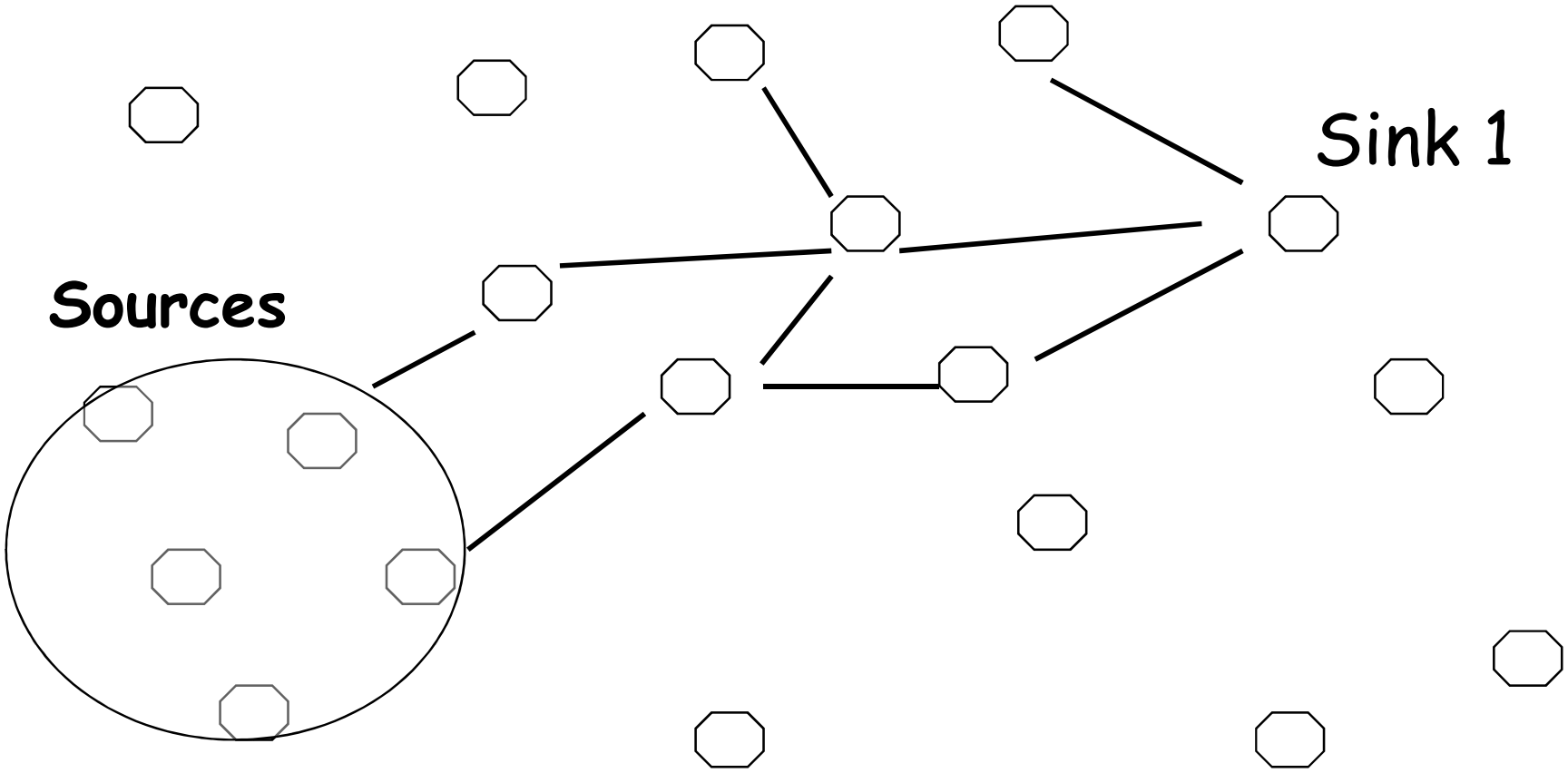
- Quand un noeud reçoit un « intérêt », il:
  - Vérifie dans le cache pour voir si une entrée est présente.
  - Si pas d'entrée, une entrée est créée avec un seul gradient (pour le voisin ayant envoyé cet « intérêt »)
  - Le Gradient spécifie la direction et le débit des données.

Interests

- Renvoyer un « intérêt » à un sous-ensemble de ses voisins
  - Cette approche est essentiellement fondée sur le flooding
  - D'autres approches possibles: probabilistes, location-based et d'autres approches d'acheminement intelligentes

- Semblable à la formation d'arbre multicast, au niveau du puit au lieu de la source

Interest Propagation



GRADIENT SETUP

- Lorsqu'un noeud capteur détecte une cible:
  - Chercher l'entrée correspondante dans le cache d'intérêts
  - Si l'entrée est trouvée, alors calculer le taux d'événement le plus élevé parmi ses gradients

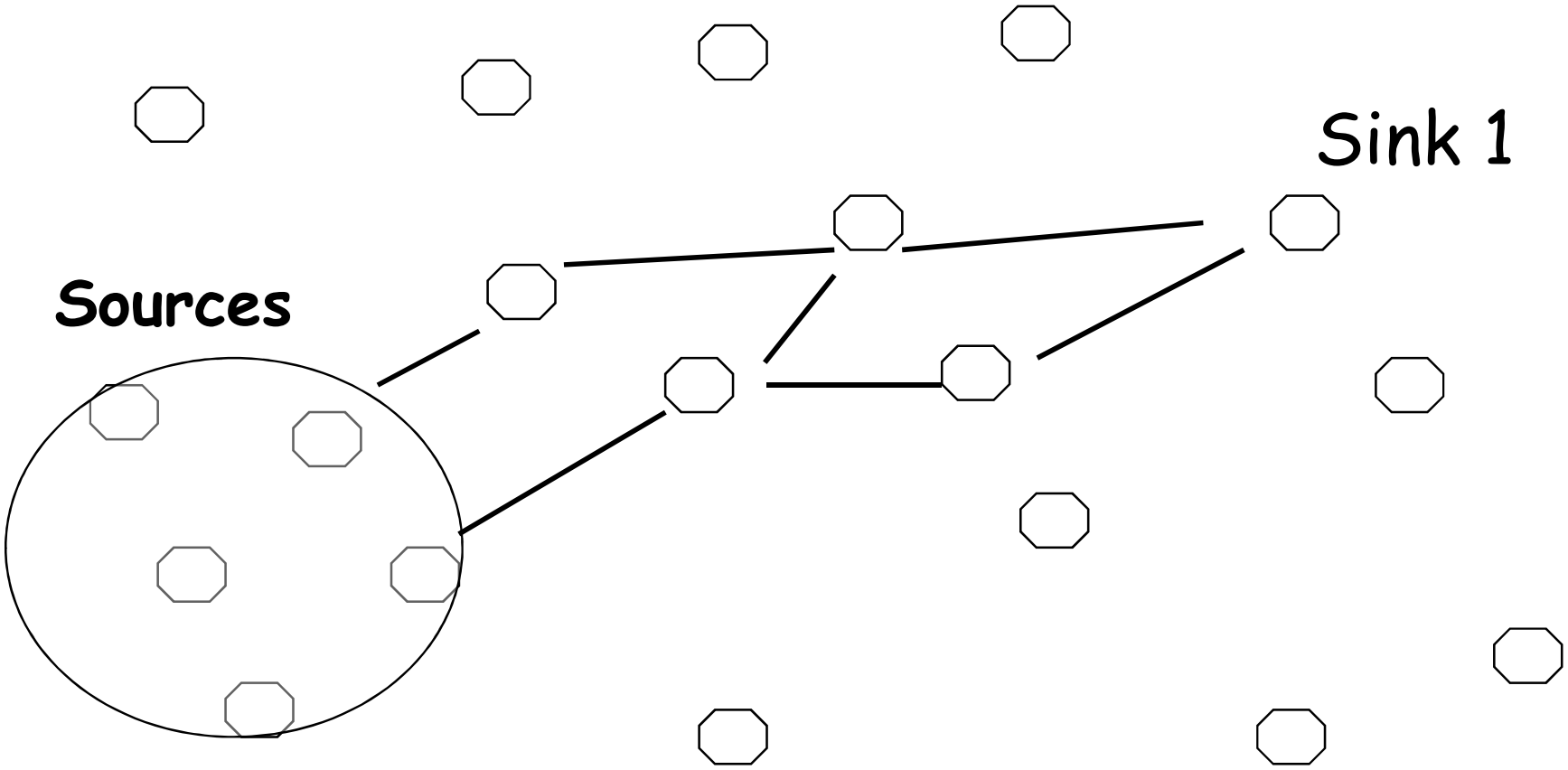
SETTING UP GRADIENTS

* Les noeuds capteurs envoient les réponses de GRADIENT SETUP en retour vers le puit.

* Chaque capteur sur le chemin compare les intérêts avec les gradients et met à jours ses champs de gradients

* Chaque capteurs achemine ces gradients vers les voisins suivants.

Gradient Setup



Règles locales pour la Propagation des Intérêts

- Juste le flooding (des intérêts)

- Techniques plus sophistiquées possibles:
  * Propagation directionnelle des intérêts fondée sur une information agrégée dans le cache

    "J'ai récemment entendu parler d'une activité suspicieuse du noeud voisin A, alors laisser moi envoyer cet intérêt pour les intrusions récentes de ce voisin"

Règles locales pour établir les Gradients

- Le gradient le plus élevé vers le voisin qui envoie l'intérêt en premier

- Autres e.x. possibles,

  * vers le voisin avec l'énergie résiduelle la plus élevée (gradients d'énergie);

  * gradients probabilistes.

Gradient Reinforcement

- Tous les gradients remontent au puit (destination/utilisateur).

- Le puit sélectionne le meilleur chemin en utilisant le contenu des paquets de gradient collectés et des besoins de l'application; c à d., le puit sélectionne un ensemble approprié de voisin(s) (meilleur lien, plus faible délai, etc.)

- En conséquence, le puit envoie, en unicast, un paquet de renforcement au prochain saut indiquant le chemin choisi (en se fondant sur le paquet de gradient)

Gradient Reinforcement

- Chaque capteur intermédiaire achemine le paquet de renforcement au prochain saut en utilisant le même principe;
  - c. à d., chaque noeud sélectionné achemine ce nouvel intérêt à un sous ensemble de ses voisins; en sélectionnant un ensemble de chemins plus petit

- A la fin, le chemin de données de la source à la destination sera établi.

Gradient Reinforcement

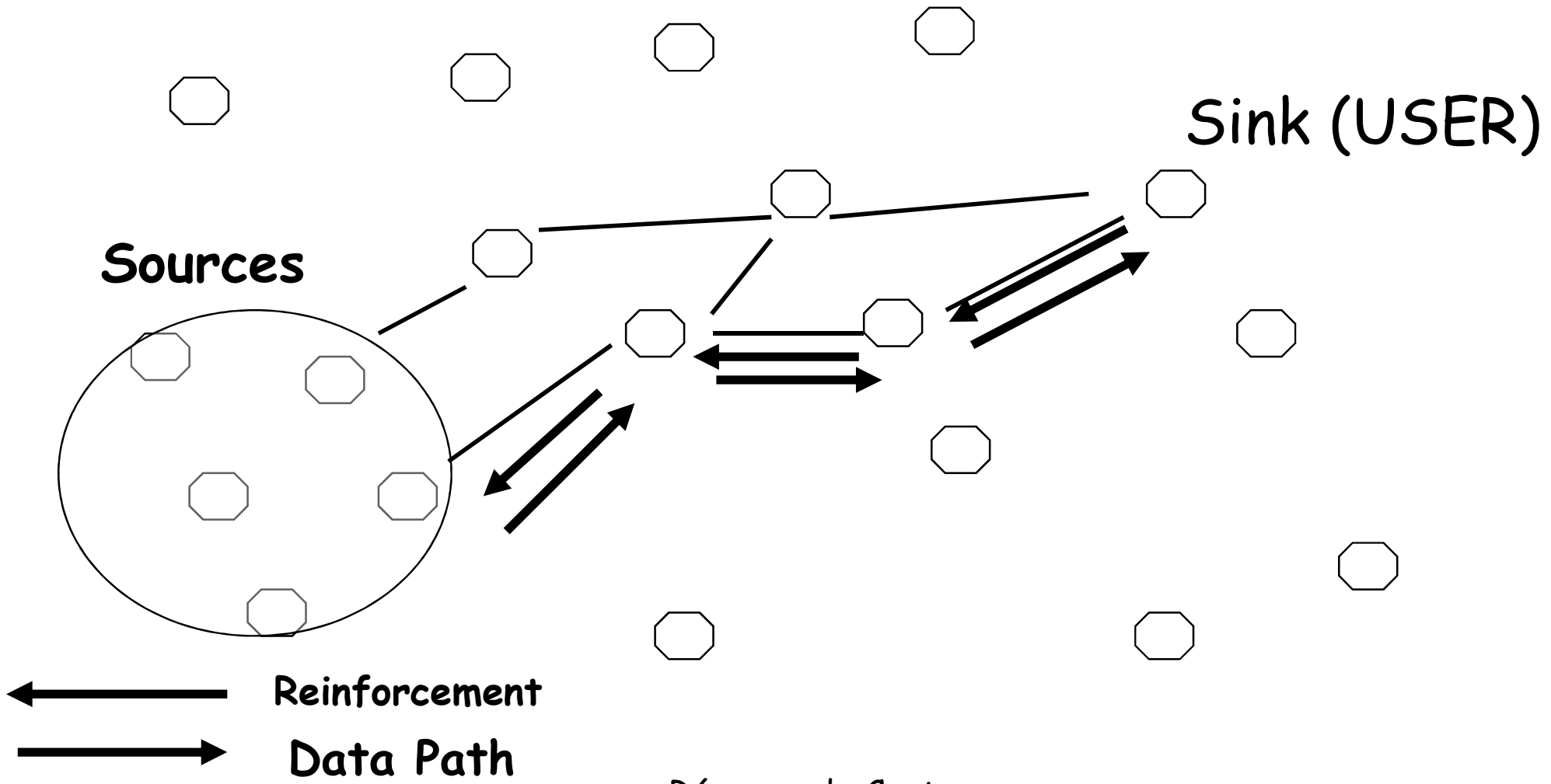
- Plusieurs critères, pour choisir le chemin qui sera renforcé, peuvent être définis
  - Quantité de données reçues d'un voisin
  - Taux de pertes
  - Variance de délai observée

Data Propagation

- Les Données seront envoyées selon le débit des sources vers le puit en utilisant le chemin de données établi.

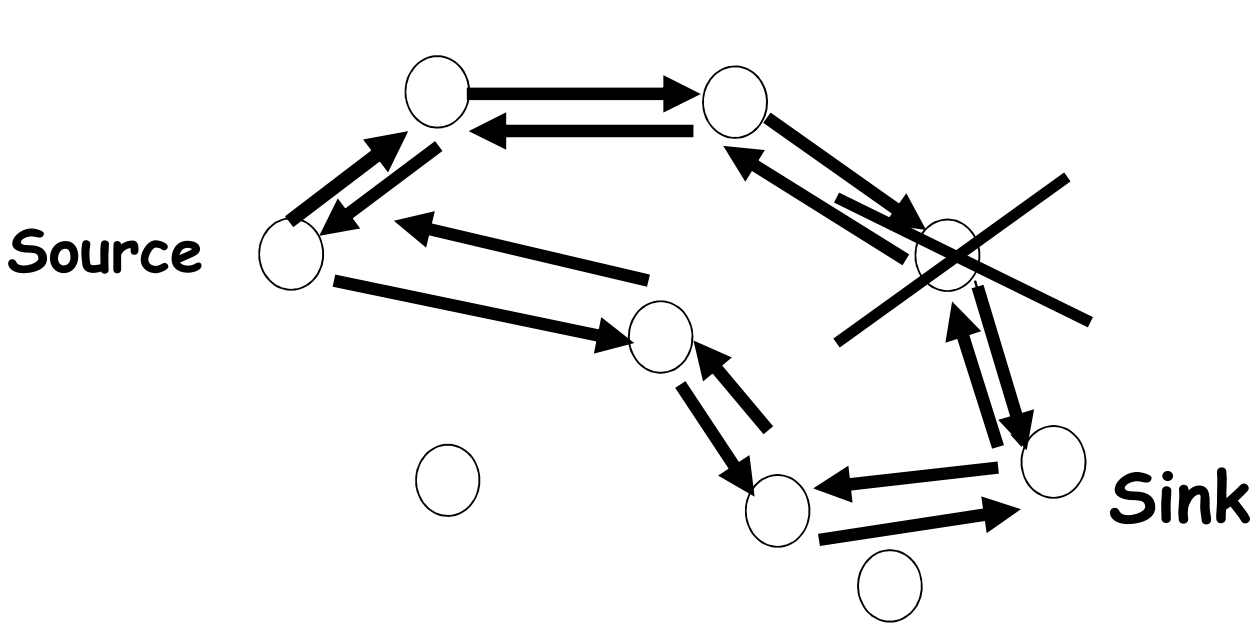
- Chaque noeud intermédiaire achemine les données à son voisin du prochain saut.

Reinforcement and Data Propagation



Negative Reinforcement

- Time out
- Dégrader explicitement un chemin en renvoyant des interêts avec des débits de données plus faibles.



→ New Data Path

Data Path

Negative Reinforcement

→ Reinforcement

Choix de Transmission de Données

- Les différentes règles locales d'acheminement peuvent permettre différents modes de transfert
  - transfert single-path
  - transfert multi-path, avec un trafic, sur chaque lien, proportionnel à son gradient
  - transfert d'une source vers de multiples puits
  - transfert de plusieurs sources vers de multiple puits

Directed Diffusion – Extensions

- Two-Phase Pull
  - Souffre de problèmes d'inondations par des "interests"
- Push Diffusion – "Data Advertisement" par les Sources
  - Le Puit envoie des paquets de « reinforcement».

Directed Diffusion vs SPIN

- Dans DD → Le puit sollicite les capteurs pour lui envoyer une donnée spécifique lorsque celle-ci est disponible en faisant de l'inondation de paquets "d'interests".

- Dans SPIN → Les capteurs annoncent la disponibilité des données permettant aux puits de les demander.

Directed Diffusion
Avantages

* DD est orienté data → pas besoin de mécanisme d'adressage de noeuds.

* Chaque noeud est supposé faire de l'agrégation, utiliser un cache, en plus de faire de la capture.

* DD est efficace sur le plan énergétique puisqu'il est à la demande et aucun besoin de maintenir un état global de la topologie.

Directed Diffusion Inconvénients

- N'est généralement pas applicable comme il est fondé sur un modèle de transfert de données dirigé par les requêtes.
- Pour les applications DYNAMIQUES nécessitant le transfert de données en continu (e.x. contrôle d'environnement) → DD n'est pas un bon choix.
- Les mécanismes de Nommage dépendent de l'application et à chaque moment doivent être définis a priori.
- Le processus de mise en correspondance entre les requêtes et les données ajoute de l'overhead au niveau des capteurs.

Comparaison entre les Algorithmes de Routage Orientés Data

Attributs / Algo.	Efficacité Données	Efficacité Énergétique (ratio data/énergie)	Complexité des états
Flooding	Le plus rapide	Rapport b/c Faible Implosion	Faible, montant
Gossiping	Le plus lent	Le plus faible (Random walk)	Aucun
SPIN	Très rapide	Plus élevé que précédemment, SPIN-EC proche de l'idéal	Paire Donnée- voisin
Directed Diffusion	Assez Rapide	Plus élevée que flooding + agrégation forte	Complexe: Voisin X Interest

Protocoles Hiérarchiques

- Les protocoles avec une architecture hiérarchique sont proposés pour répondre aux problèmes de *passage à l'échelle* et de *consommation d'énergie* des réseaux de capteurs.

- Les noeuds capteurs forment des clusters ou les cluster-heads agrégent et fusionnent les données pour conserver l'énergie.
- Les cluster-heads peuvent former, entre eux, un autre niveau de clusters avant d'atteindre le puit.

Protocoles Hiérarchiques

- Low-Energy Adaptive Clustering Hierarchy (LEACH) (Heinzelman'00)
- Power-efficient GAthering in Sensor Information Systems (PEGASIS) (Lindsey01, Lindsey02),
- Threshold sensitive Energy Efficient sensor Network protocol (TEEN) (Manjeshwar01)
- AdaPtive Threshold sensitive Energy Efficient sensor Network protocol (APTEEN) (Manjeshwar02).

Low Energy Adaptive Clustering Hierarchy

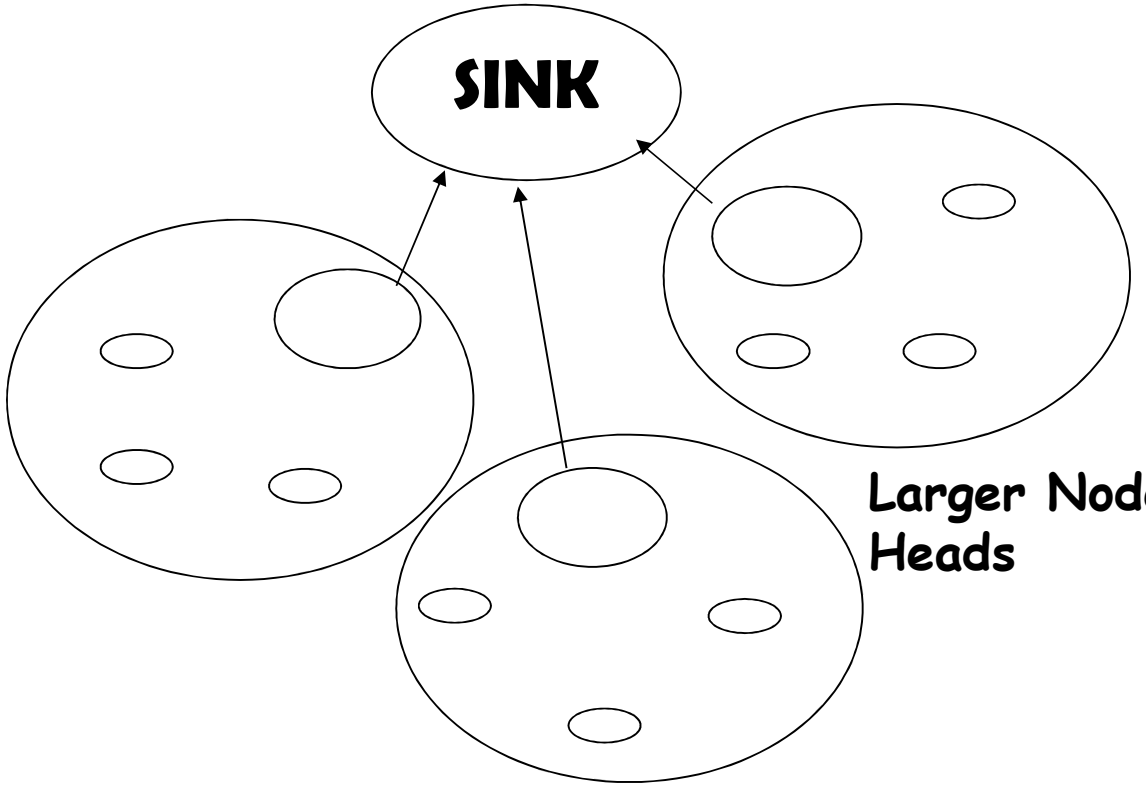
W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks,'' *IEEE Proc. of the Hawaii Int. Conf. on System Sciences*, pp. 1-10, January, 2000. Longer version in IEEE Tr. on Wireless Com., pp.660-670, Oct. 2002

LEACH is a clustering based protocol which minimizes energy dissipation in sensor networks.

Idea:

* Randomly select sensor nodes as cluster heads, so the high energy dissipation in communicating with the base station is spread to all sensor nodes in the network.
* Forming clusters is based on the received signal strength.
* Cluster heads can then be used kind of routers (relays) to the sink.

LEACH



Larger Nodes denote Cluster Heads

LEACH

Two Phases: Set-up Phase and Steady-Phase

<u>Set-up Phase:</u>

* Sensors may elect themselves to be a local cluster head at any time with a certain probability.
  (Reason: to balance the energy dissipation)
* A sensor node chooses a random number between 0 and 1

LEACH

- If this random number is less than the threshold T(n), the sensor node becomes a cluster-head.

$$T(n) = P / \{1 - P[r \bmod (1/P)]\} \quad \text{if } n \text{ is element of } G$$

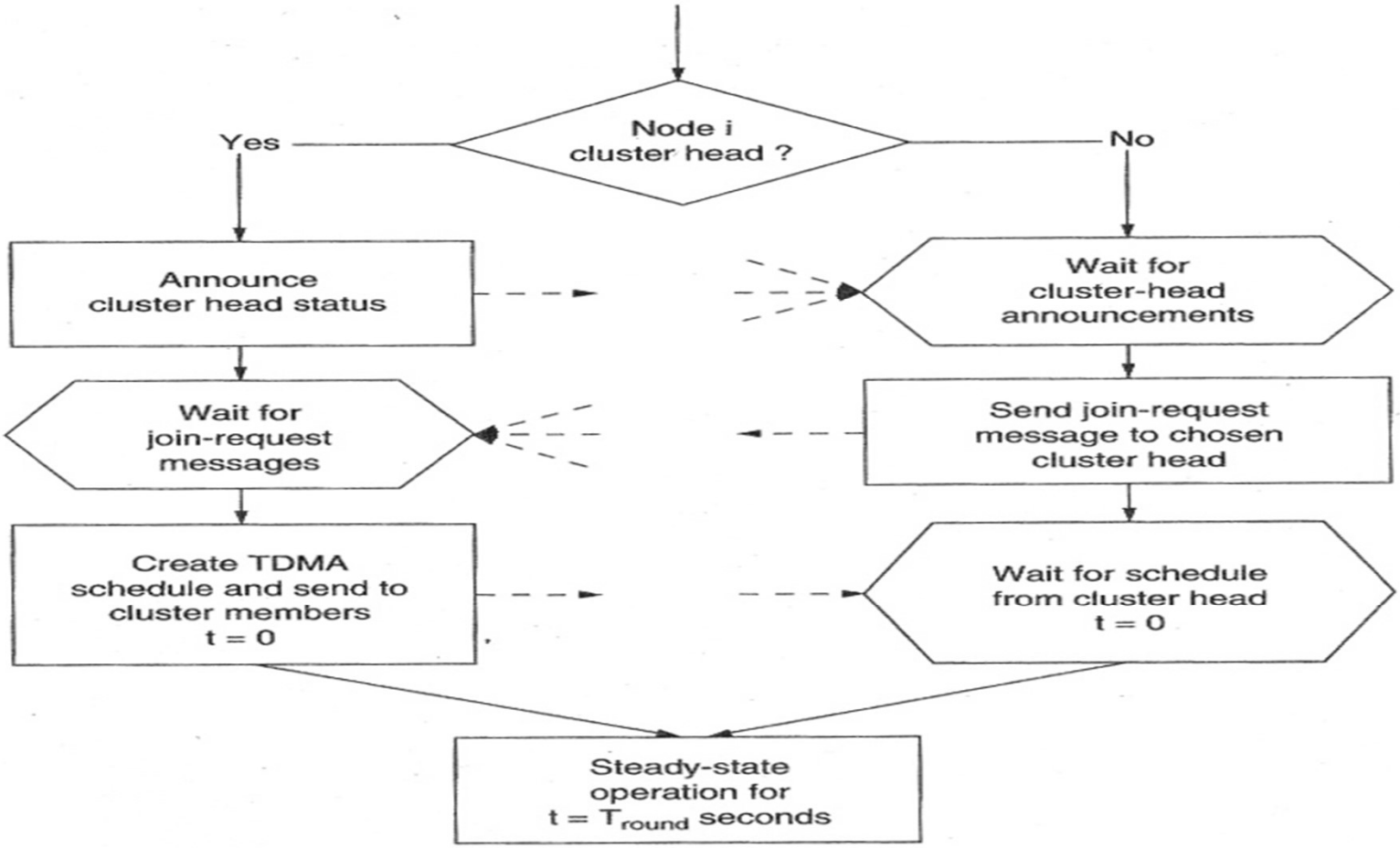
$$T(n) = 0 \quad \text{otherwise}$$

where P is the desired percentage to become a cluster head (e.g., 0.05)

r is the current round

G is the set of nodes that have not been a cluster head in the last 1/P rounds.

CLUSTER HEAD SELECTION



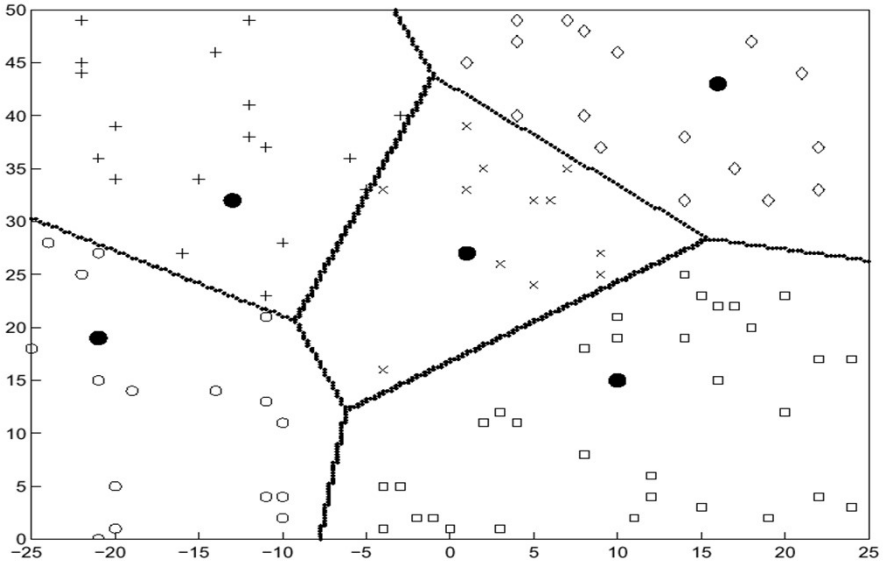
Flowchart of cluster head election in LEACH protocol.

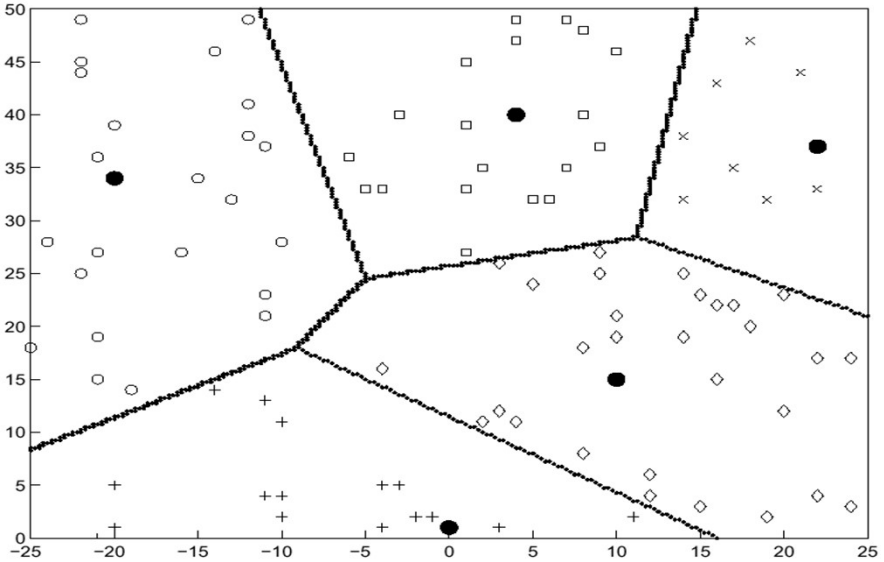
LEACH

- After the cluster heads are selected, the cluster heads advertise to all sensor nodes in the network that they are the new cluster heads.

- Each node accesses the network through the cluster head that requires minimum energy to reach.

Dynamic Clusters





LEACH

- Once the nodes receive the advertisement, they determine the cluster that they want to belong based on the signal strength of the advertisement from the cluster heads to the sensor nodes.

- The nodes inform the appropriate cluster heads that they will be a member of the cluster.

- Afterwards the cluster heads assign the time on which the sensor nodes can send data to them.

LEACH

STEADY STATE PHASE:

- Sensors begin to sense and transmit data to the cluster heads which aggregate data from the nodes in their clusters.

- After a certain period of time spent on the steady state, the network goes into start-up phase again and enters another round of selecting cluster heads.

LEACH

- Optimum Number of Clusters ---????????

  - too few: nodes far from cluster heads
  - too many: many nodes send data to SINK.

LEACH - CONCLUSIONS

- Achieves over a factor of 7 reduction in energy dissipation compared to direct communication.
- The nodes die randomly and dynamic clustering increases the lifetime of the system.
- It is completely distributed and requires no global knowledge of the network.

LEACH - CONCLUSIONS

- It is not applicable to networks deployed in large regions.
- Furthermore, the idea of dynamic clustering brings extra overhead, e.g., head changes, advertisements etc. which may diminish the gain in energy consumption.

Location Based (Geographic) Routing Protocols

- Routing tables contain information to which next hop a packet should be forwarded
  - Explicitly constructed
- Alternative: Implicitly infer this information from physical placement of nodes
  - Position of current node, current neighbors, destination known – send to a neighbor in the right direction as next hop

  → *Geographic routing*
- Options
  - Send to any node in a given area – *geocasting*
  - Use position information to aid in routing – *position-based routing*
    - Might need a location service to map node ID to node position

Location-Based (Geographical Routing) Protocols

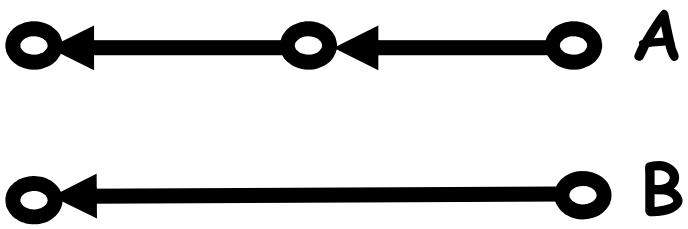
1. MECN and SMECN (Li01)
2. PRADA (Melodia/Pompili/Akyildiz'05)

Minimum Energy Communication Network

L. Li and J.Y. Halpern, "Minimum-Energy Mobile Wireless Networks Revisited". Proc. of IEEE Int. Conf. on Communications (ICC'01), Helsinki, Finland, June 2001.

OVERVIEW:

* Each node knows its exact location through GPS
* Network is represented by a graph G', and it is assumed that the resulting graph is connected
* A sub-graph G of G' is computed.
* G connects all nodes with minimum energy cost.



Connection A requires less energy than connection B because the power required to transmit between a pair of nodes increases as the n^{th} power of the distance between them (n>=2).

Minimum Energy Communication Network

Main Idea:

- Find a sub-network which will have less number of nodes and will require less power for transmission between any two particular nodes.
- In this way, global minimum power paths are found without considering all the nodes in the network.
- This is performed using a localized search for each node considering its relay region.

Minimum Energy Communication Network

- The protocol details:
  - It takes the positions of a two dimensional plane and constructs an enclosure graph, which consists of all the enclosures of each transmit node in the graph.
  - This construction requires local computations in the nodes.
  - Finds optimal links on the enclosure graph.
  - It uses distributed Belmann-Ford shortest path algorithm with power consumption as the cost metric.

Geographical Routing: Minimum Energy Communication Network

- A minimum power topology for fixed sensor nodes is determined
- MECN identifies a relay region for every node.
- The relay region consists of nodes in a surrounding area where transmitting through those nodes is more energy efficient than direct transmission.

- The relay region for node pair (i, r) is depicted in the Figure:
- The enclosure of a node i is then created by taking the union of all relay regions that node i can reach.

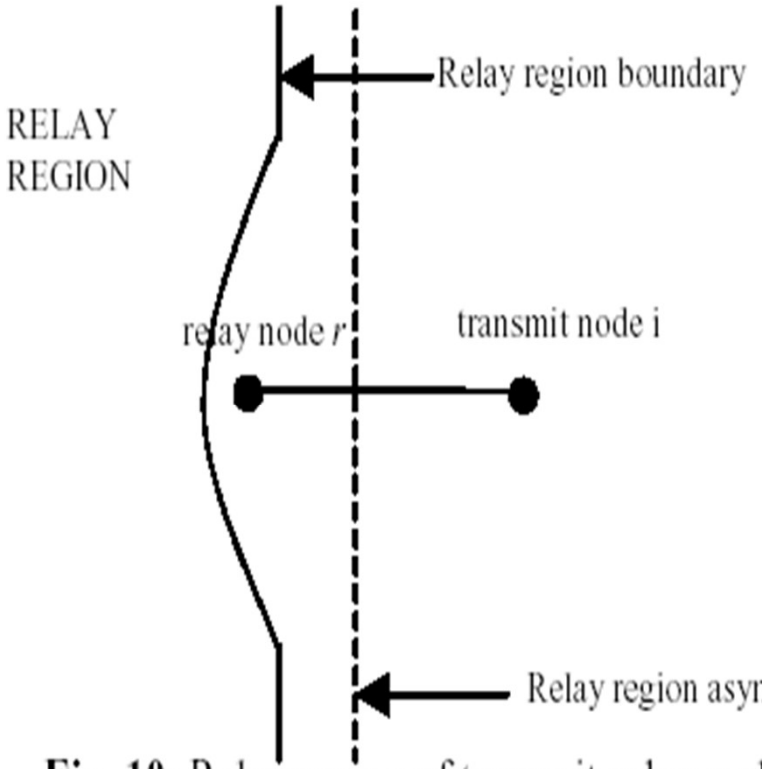


Fig. 10: Relay region of transmit-relay node pair (i, r) in MECN.

Distributed Topology Control

T. Melodia, D. Pompili, I.F. Akyildiz,
IEEE JSAC (Journal of Selected Areas in Communications), March 2005.

Primary design constraints for sensor network protocols:

Energy Efficiency

- Optimize energy consumption of protocols at each layer

AND

- Cross-Layer approach: jointly optimize networking functionalities o different layers

Scalability

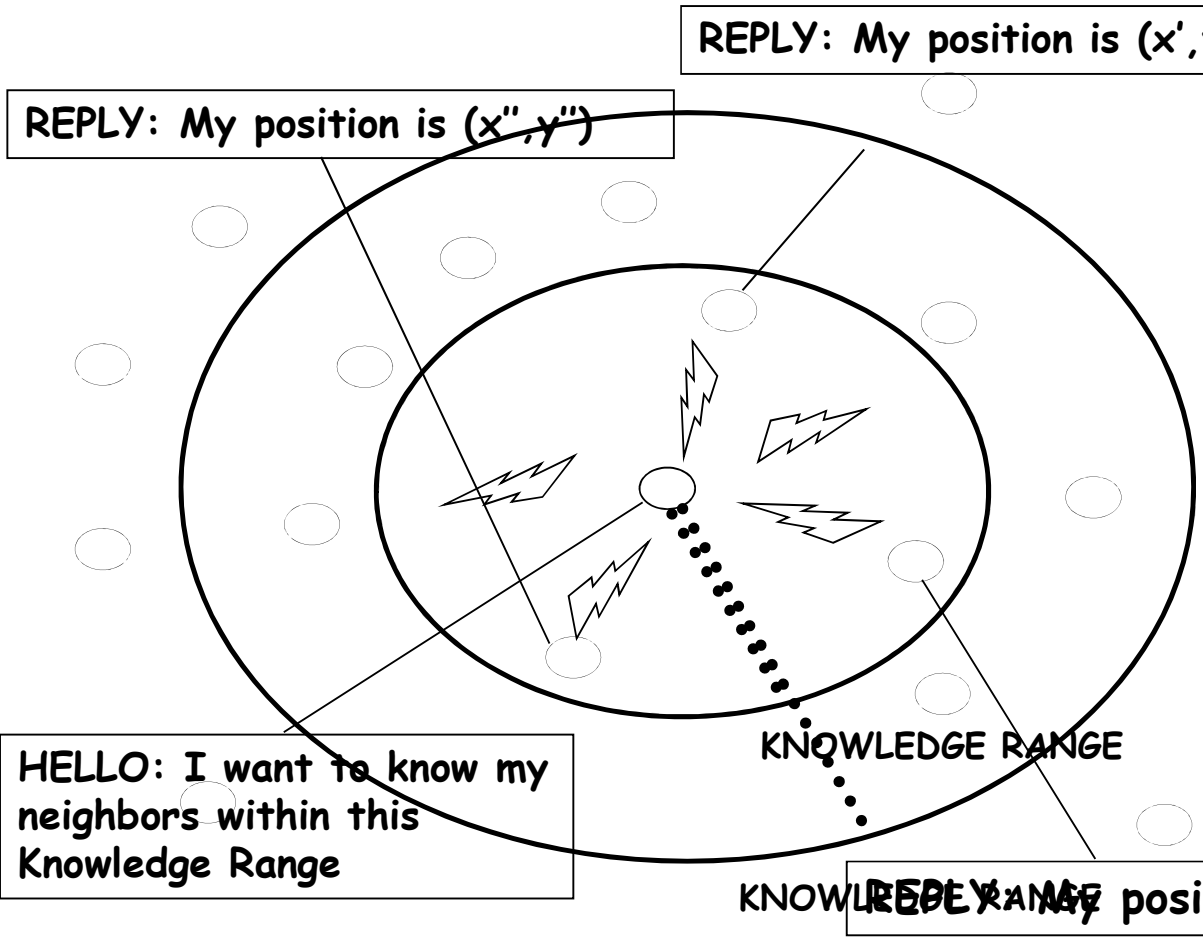
- Performance should not be worsened with increasing number of nodes
- Scalable routing solution is crucial

Motivation

- We consider interactions between
  - Energy Efficiency of geographical routing
  - Topology Control:
    - Given a node in a wireless setting, what are its neighbors, i.e., the nodes it can reach in one hop?
      The number of neighbors can be adjusted with "power control"

Can we improve the Energy Efficiency of Geographical Routing with a cross layer approach by taking into account Topology Control?

Geographical Routing - Basics



Topology Discovery

Discover the position of neighbors

Short Topology Knowledge Range:

1) Small Number of close Neighbors

2) Lower Energy Consumption

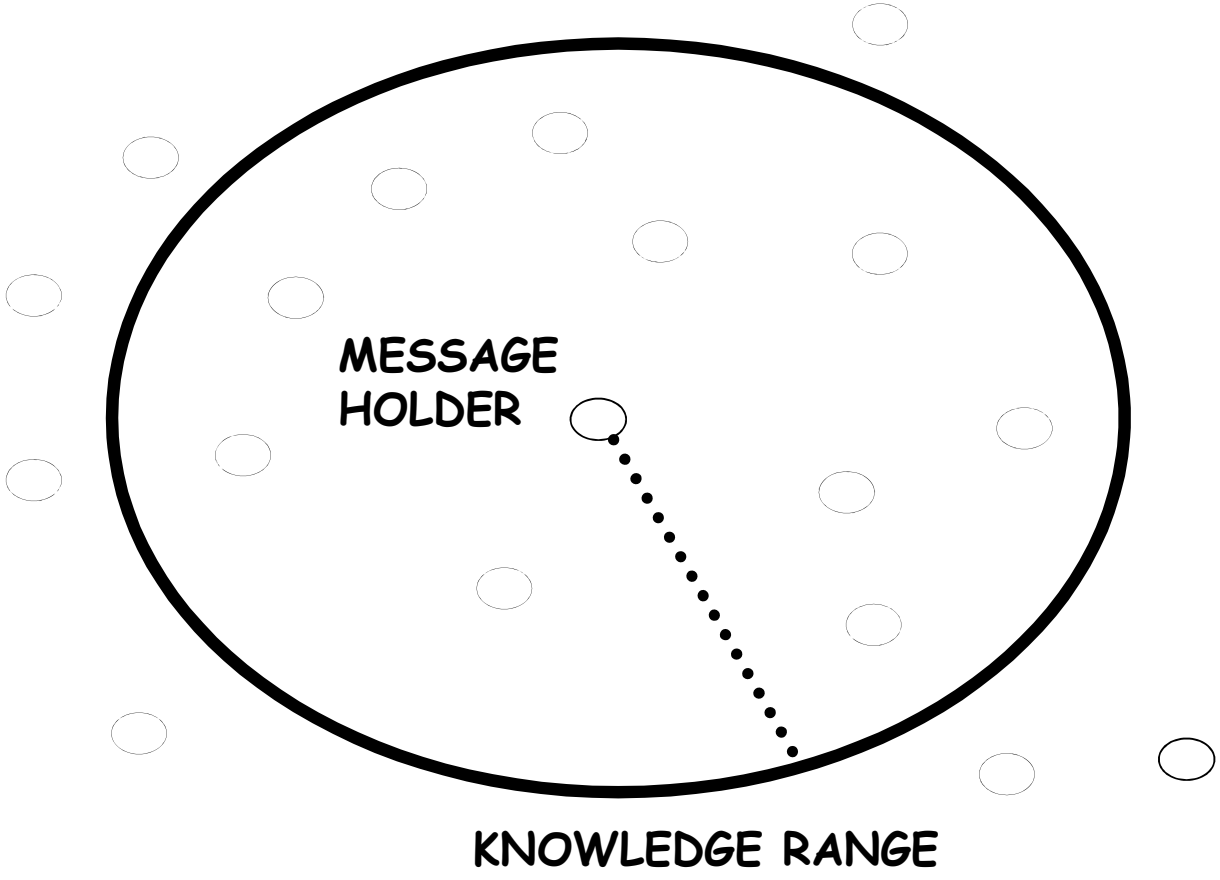
Long Topology Knowledge Range

1) Large number of Neighbors

2) Higher Energy Consumption

- In geographical routing algorithms, devices forward packets based on neighborhood information (topology information)

Geographical Routing - Basics



Next Hop Selection

Given a DESTINATION, the node that is holding the message selects the next hop according to

1) Its own position

2) The position of the destination node

3) The position of its neighbors (nodes in the Knowledge Range)

DIFFERENT FORWARDING RULES ARE POSSIBLE!

Geographical Routing - Basics

Two different notions of cost are defined

- Cost of Information:
  - the energy needed to acquire topology information, i.e., the energy necessary to exchange the associated signaling traffic
- Cost of Communication:
  - the energy needed to transmit data from source to destination on a given path in the network

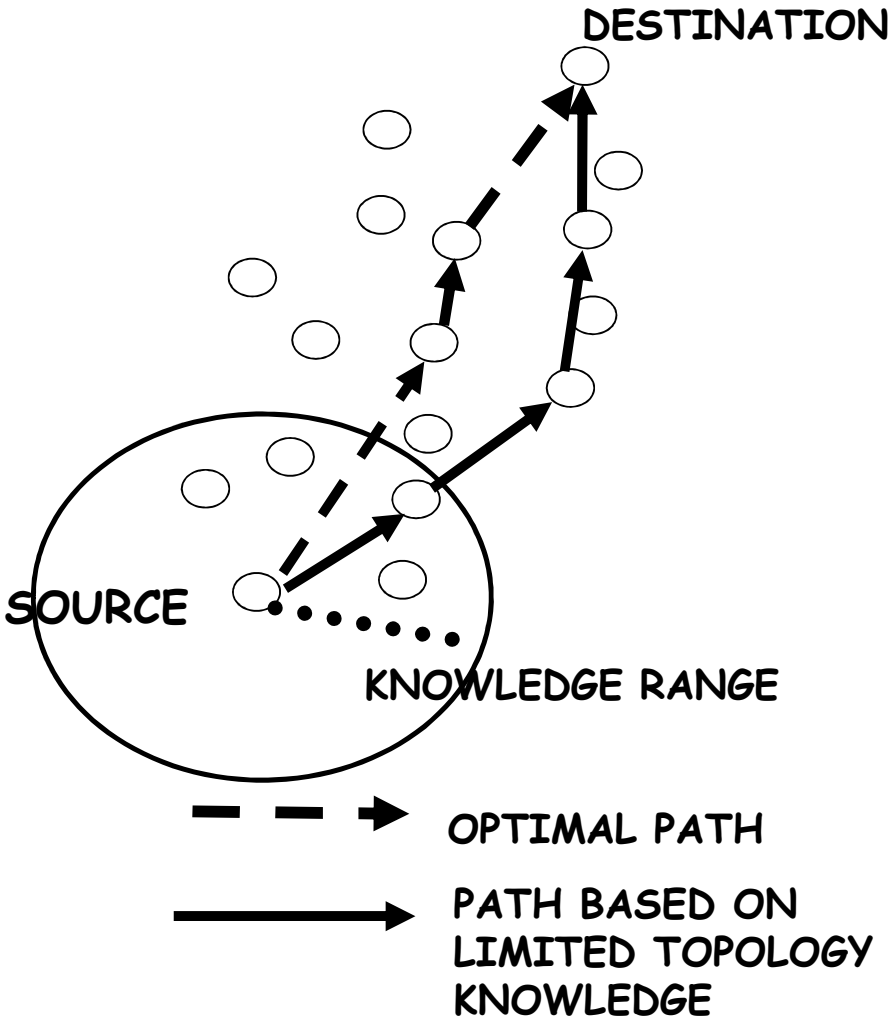
- With complete knowledge of the topology each sensor can compute the "globally" optimal path (i.e., minimum energy path)
- This way, the cost of communication is minimum but the cost of topology information is maximum

Cost of Communication vs Cost of Information

- There is a tradeoff between:
  - Cost of topology information, increases with the Knowledge Range
  - Cost of communication, usually decreases when the Knowledge Range increases

- The question we try to answer is:
  - "How extensive should be the Local Knowledge Range of the global topology in each sensor node, so that an energy efficient geographical routing can be guaranteed?"

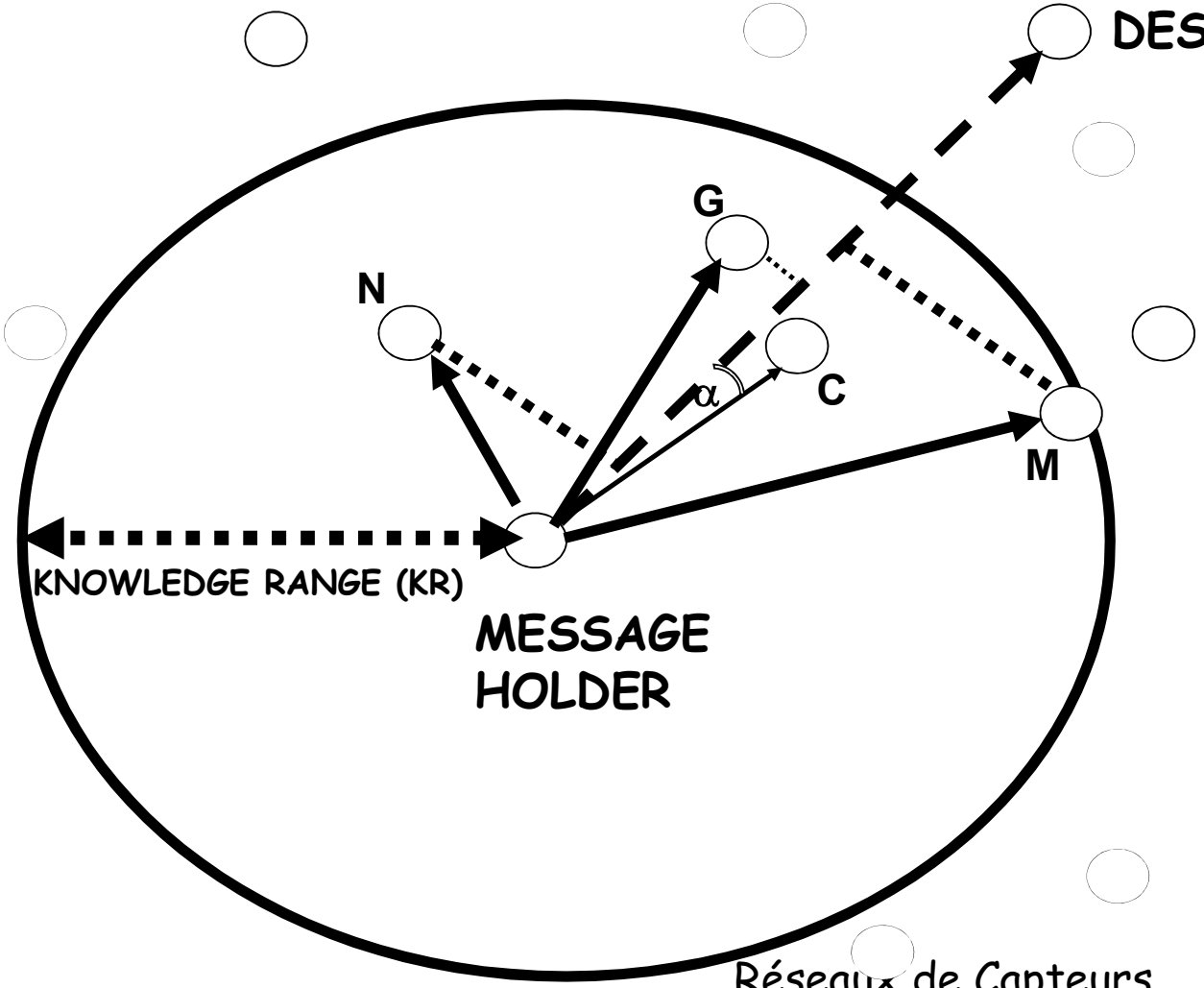
Cost of Communication vs Cost of Information



- Complete topology knowledge
  - Each device can calculate the globally optimal path
  - High cost to acquire topology information (signalling)
- Limited topology knowledge
  - can lead to suboptimal paths
  - Cost of topology information depends on the *Knowledge Range (KR)*

Which path is better if we also take the Cost of Information into account?

Geographical Routing: Existing Greedy Forwarding Schemes



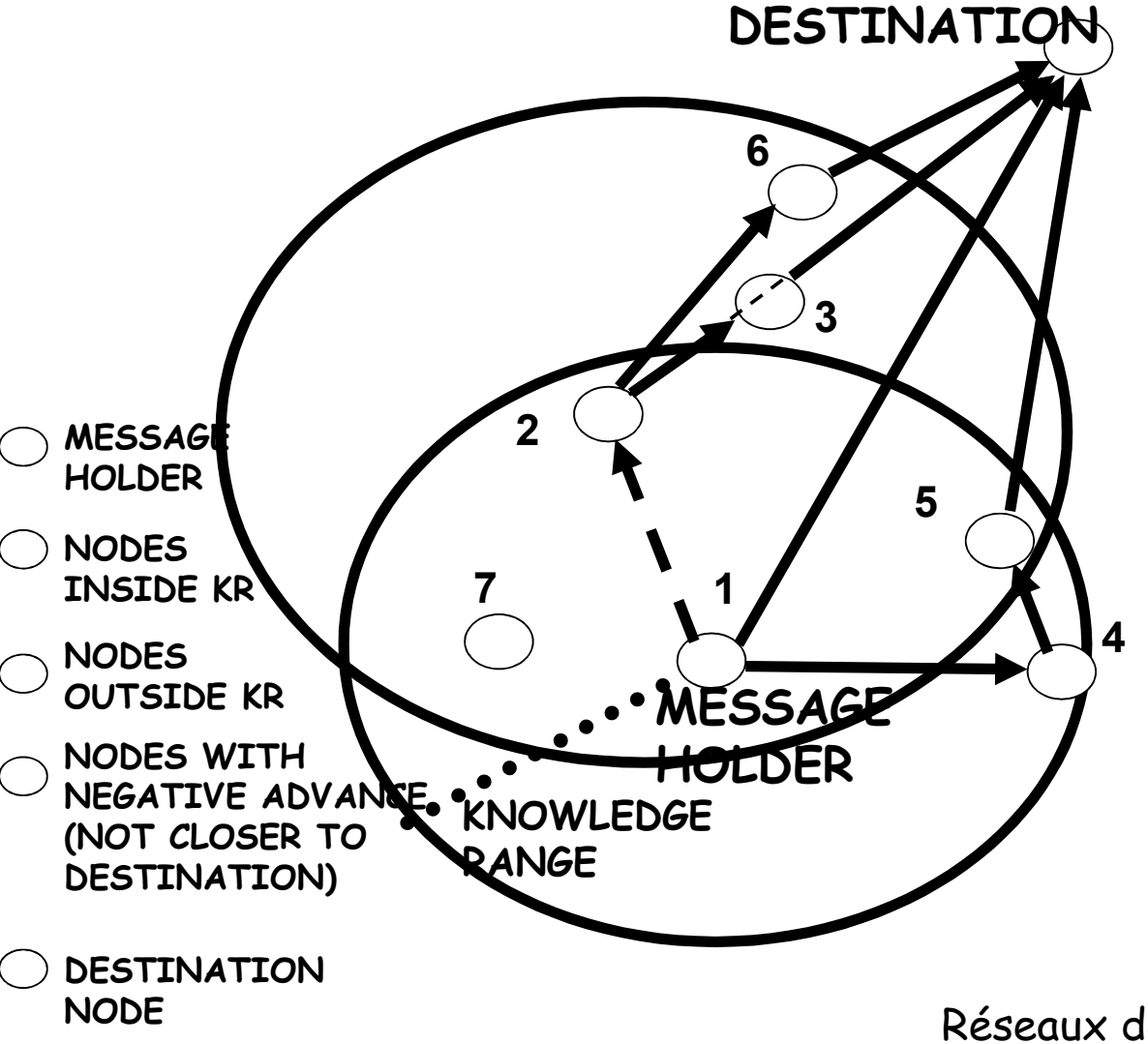
M: Most Forward within Radius (MFR)

N: Nearest Forward Progress (NFP)

G: Greedy Routing Scheme (GRS)

C: Compass

PTKF: Partial Topology Knowledge Forwarding
(Melodia/Pompili/Akyildiz'05)



- Node 1 chooses the optimal path according to its Knowledge Range
- Optimal path is minimum energy path
- Only nodes with positive advance are feasible next hops
- If "yellow" path is selected, the packet is forwarded to node 2
- Node 2 calculates the optimal path according to its own (different) Knowledge Range
- Node 6 is now on the optimal path and receives the packet
- The packet is forwarded to the destination
- Unlike greedy forwarding schemes, PTKF selects the next hop based on energy consumption!

Topology Knowledge Range Problem

Total number of nodes

$$C^*(\underline{R}^*) = \min_{\underline{R}} \left\{ \sum_{k=1}^{N} \left[C_k^{INF}(r_k) + C_k^{COM}(\underline{R}) \right] \right\}$$

$C_k^{INF}(r_k)$ **Cost of Information:** energy needed for node k to acquire topology information with a knowledge range r_k

$C_k^{COM}(\underline{R})$ **Cost of Communication:** energy needed by node k to transmit user data given:

· the vector of knowledge ranges: $\underline{R} = [r_1, r_2, .., r_N]$

OBJECTIVE: Find the OPTIMAL VECTOR $\underline{R^*}$ of Knowledge Ranges that minimizes the overall cost of the network

Compare different forwarding rules

Energy Model

$$c_{ij} = \underbrace{2 \cdot E_{elec}}_{\text{DISTANCE INDEPENDENT FACTOR}} + \underbrace{E_{amp} \cdot d_{ij}^{\alpha}}_{\text{DISTANCE DEPENDENT FACTOR}}$$

- Energy needed to transmit one bit between nodes i and j
  - $2 \leq \alpha \leq 5$
  - E_{elec}: the energy needed by the transmitter [receiver] to transmit [receive] one bit
  - E_{elec}: [Joule/bit]
  - E_{amp} : [Joule/bit/m^{α}]
- When E_{elec} is low as compared to $E_{amp} \cdot d_{ij}^{\alpha}$, multi hop paths are more energy efficient!

ILP Formulation

- Integer Linear Programming Formulation
- We find the optimal solution given the mathematical model of the problem and input data.
- We use discrete values of the knowledge ranges KR:
  - Example: [0, 5, 10, 15, 20, 25] m

ILP Formulation

- Based on binary variables:

Given

$f_{kd}^{ij} = 1$ iff j is the next hop of i towards the destination d with the k-th Knowledge Range (FORWARDING RULE)

$a_{im}(k) = 1$ iff node m is a neighbor of node i with k-th KR

Unknown

(found by the solver)

$y_i^k = 1$ iff node i selects the k-th KR

$x_{ij}^{sd} = 1$ iff connection between source s and destination d uses the link between node i and node j, according to the rule selected

ILP Formulation

Minimize:

$$C^{TOT} = \sum_{i \in V} \left(C_i^{COM} + C_i^{INF} \right)$$

Set of Nodes

Subject to the following constraints:

Cost of Information for node i:

Number of Neighbors of node i with k-th KR

$$C_i^{INF} = L_N \cdot E_{amp} \cdot \sum_{k \in R} y_i^k \cdot r^{\alpha}(k) + \left(\sum_{k \in R} (y_i^k \cdot N_i(k)) + 1 \right) \cdot L_N \cdot E_{elec} +$$

Length of the HELLO message (bits)

Energy needed to transmit HELLO message with Knowledge Range k

Energy needed for the neighbors to receive the HELLO message

$$+ \sum_{m \in N_i(k)} \left(L_U \cdot E_{amp} \cdot d_{mi}^{\alpha} + 2 \cdot L_U \cdot E_{elec} \right) \cdot \sum_{k \in R} y_i^k \cdot a_{im}(k) \Big) \cdot \frac{1}{T_M}, \quad \forall i \in V$$

Length of the REPLY message (bits)

Energy needed for the neighbors to send their REPLY message

Period between two consecutive HELLO messages

ILP Formulation

Constraint on cost of communication for node i:

$$C_i^{COM} = \sum_{s \in S} \sum_{d \in D} \sum_{j \in V} \left(x_{ij}^{sd} \cdot p_i^{sd} \cdot \left(2 \cdot E_{elec} + E_{amp} \cdot d_{ij}^{\alpha} \right) \right), \quad \forall i \in V$$

Set of Source nodes

Set of Destination nodes

Data rate of the connection between s and d (0= no connection)

For all source-destination pairs, for each node j (possibly neighbor of i), if the connection between s and d passes through the link (i,j)

($x_{ij}^{sd} = 1$), then the formula includes the cost of the link (i,j)

($2 \cdot E_{elec} + E_{amp} \cdot d_{ij}^{\alpha}$) in the cost of communication for node i

ILP Formulation

- Impose that paths are built according to the forwarding rule expressed by the f variables

$$x_{ij}^{sd} \le \sum_{k \in R} \left(y_i^k \cdot f_{dk}^{ij} \right), \ \forall s \in S, \ \forall d \in D, \ \forall i, j \in V$$

$$x_{sj}^{sd} = \sum_{k \in R} \left(y_s^k \cdot f_{dk}^{sj} \right), \ \forall s \in S, \ \forall d \in D, \ \forall j \in V \ s.t. \ s \neq d$$

- Imposes that each node selects only one KR

$$\sum_{k \in R} y_i^k = 1, \ \forall i$$

ILP Formulation

3 FLOW CONSERVATION CONSTRAINTS

- Each source generates one flow

$$\sum_{j \in V} \left(x_{sj}^{sd} - x_{js}^{sd} \right) = 1, \forall s \in S, \forall d \in D, s.t. s \neq d$$

- Each destination receives one flow

$$\sum_{j \in V} \left(x_{dj}^{sd} - x_{jd}^{sd} \right) = -1, \forall s \in S, \forall d \in D, s.t. s \neq d$$

- Each other node retransmits exactly the flows that it receives

$$\sum_{j \in V} \left(x_{ij}^{sd} - x_{ji}^{sd} \right) = 0, \forall s \in S, \forall d \in D, \forall i \in V \; s.t. s \neq d, i \neq s, i \neq d$$

Online Solution: PRADA

- ILP problem: can be solved with a centralized algorithm
- ILP is NP-complete-> Limited number of nodes

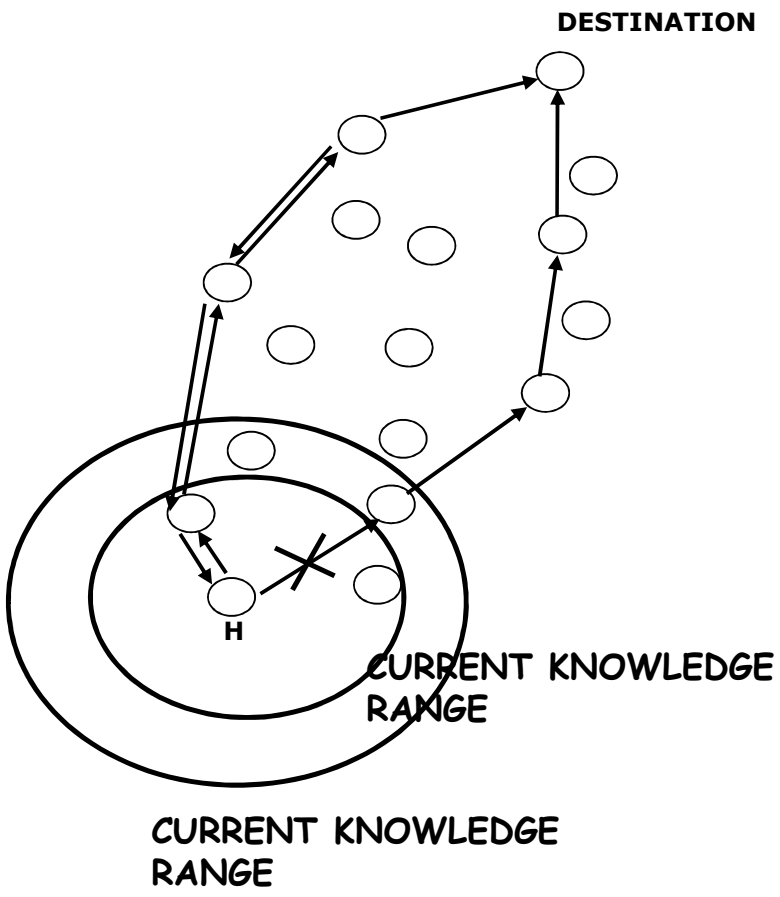
- For solution of the problem, PRADA was developed :
  - A PRobe bAsed Distributed algorithm for knowledge rAnge adjustment

Online Solution: PRADA

- PRADA allows nodes to distributively select their Knowledge Range by means of feedback information from the network

- The Optimal Solution is used as a comparison for the performance of PRADA, whose performance should be as close as possible to the optimum

Example: PRADA Operation



- Node H sends data according to its KR (current)
- KR probe is selected
- New tentative next hop selected and probe packet is sent
- If the node knows the cost of the path to destination, it replies, otherwise the packet is forwarded
- When the packet reaches a node which has updated information for that destination, it is sent back
- Worst case: one-hop neighbor of the destination
- Done for all destinations if needed
- If the cost with the new KR is greater than with the current KR, nothing is done
- Otherwise, the new KR is selected

Pseudocode for PRADA

Algorithm 1 PRADA

begin
randomly select $r_{probe} \neq r_{current}$
for each $p_k \in \mathcal{P}_i$ **do**
  $v_i \rightarrow l^{\mathcal{F}}_{v_i}(v^k_d, r_{probe})$: probe packet
end for
wait for return packets
$C^{TOT}_i(r_{probe}) = C^{INF}_i(r_{probe}) + \sum_{p \in P_i} C^{\mathcal{P}}_i(r_{probe})$
if $(C^{TOT}_i(r_{probe}) < C^{TOT}_i(r_{current}))$ **then**
  $r_{current} = r_{probe}$
end if
end

For each connection node i is involved in

Send a probe packet to the next hop

Calculate cost associated to r_probe

If cost(r_probe)< cost(r_current)

Update r_current

NETWORK LAYER RESEARCH NEEDS

- New Addressing Mechanisms
- Store and Forward Technique that combines data fusion and aggregation.
- Routing for Mobile Sensors
  Investigate multi-hop routing techniques for high mobility environments.
- Priority Routing
  Design routing techniques that allow different priority of data to be aggregated, fused, and relayed.
- 3D Routing