Data Aggregation in Sensor Networks

Alberto Marchetti-Spaccamela

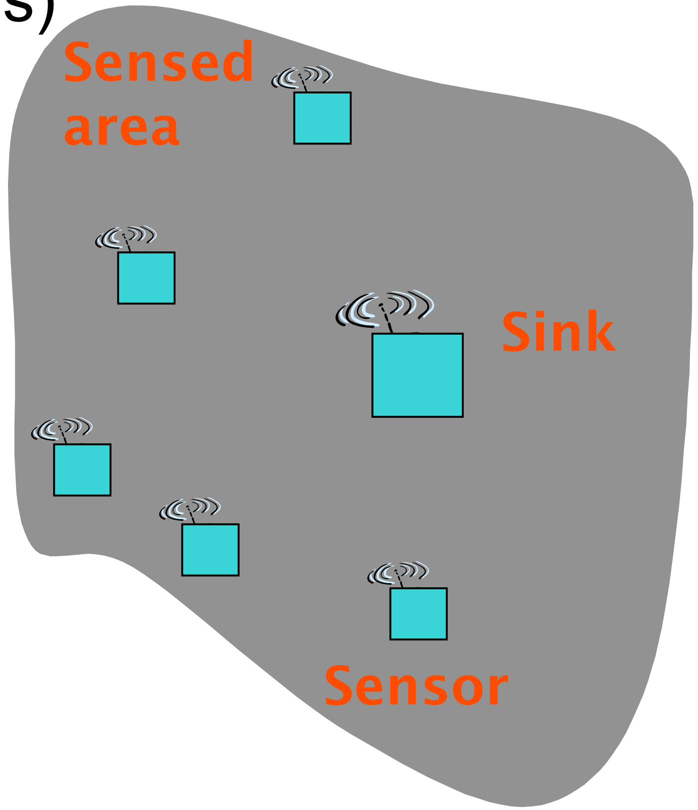
U. Rome "La Sapienza"

Outline

- Introduction and motivation
- Maximum Lifetime Data Aggregation
- Latency Constrained Data Aggregation

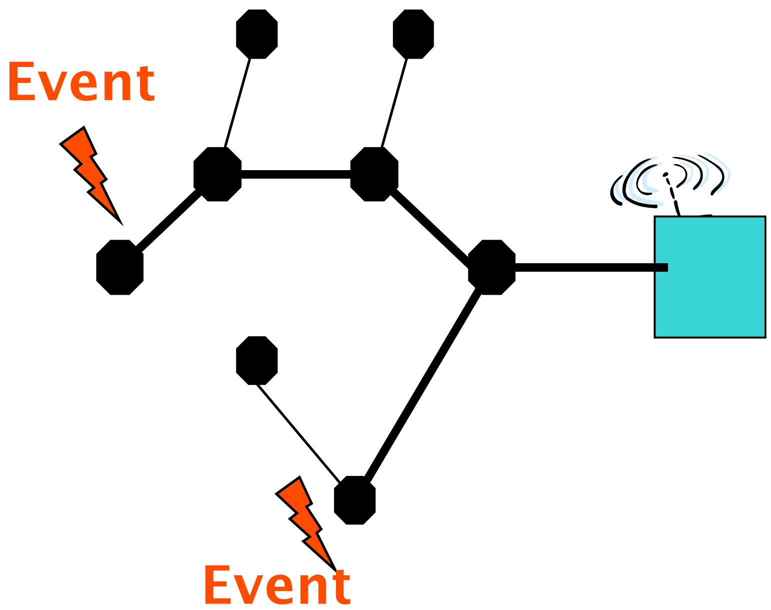
Sensor Networks

- Sensors networks are distributed based systems with source nodes (sensors) publishing data to sink node(s)
- Sensors:
  - integrated sensors
  - data processing capabilities
  - short-range radio communications
- Sink
  - possibly more powerful node
  - collects data coming from sensors



Routing in Sensor Networks

- data are collected at sensors and delivered to sink
- underlying network topology: might change dynamically



Energy Saving

- energy resources are scarce
  - Battery operated; no recharge
- most important problem in sensor networks:
  - google scholar: <energy aware sensors networks> gives 13700 articles
- most energy spent for transmitting and receiving

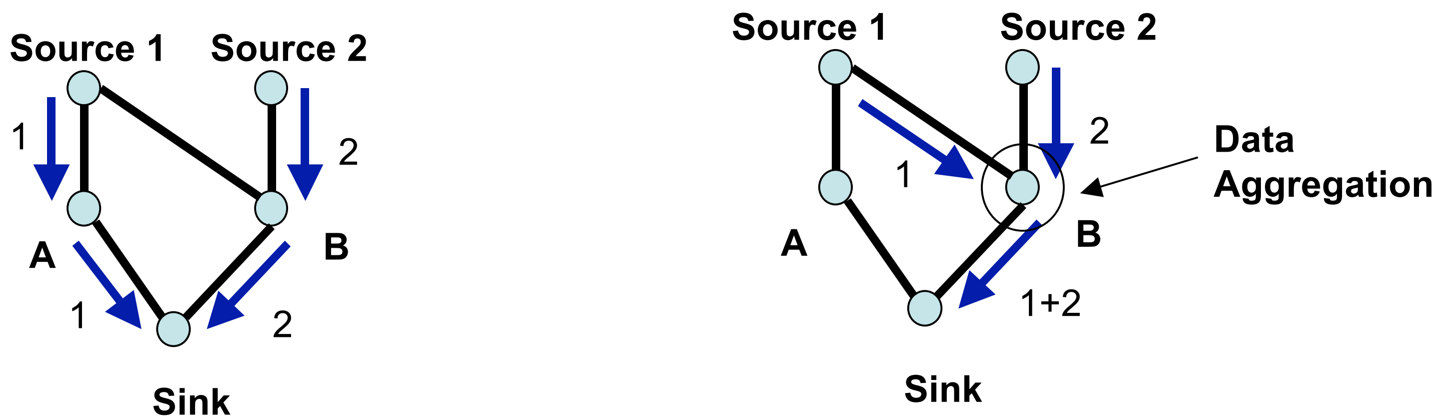
  trans. cost ≈ no. transmitted bits
- techniques
  - energy aware routing
  - data fusion
  - wake-asleep duty cycles

Energy-Efficient Sensor Networks

- Energy-aware routing protocols [Singh et al 1998]
- LEACH [Heinzelman et al 1999]
  - Clustering-based protocol for transmitting data to the base station
- Chang and Tassiulas [2001]
  - Routing algorithms that maximize the time until the sensor energies drain out
- Bharadwaj et al [2001]
  - bounds on the lifetime of an energy-constrained sensor network
- PEGASIS [Lindsey et al 2001]
  - Chains formed among sensors to gather and aggregate data
  - Sensors take turns to transmit to the base station
- PEGASIS-based hierarchical scheme [Lindsey et al 2001]
  - Reduces the delay incurred in each round of data gathering
- TinyOS [Madden et al 2002]
  - Implements basic database predicates (e.g. COUNT, MIN, MAX,AVERAGE) useful to the in-network regime.

Data Aggregation Process

- Energy savings is obtained by allowing in-network aggregation of redundant information
- A data fusion node collects results from multiple nodes
  - Less packet transmissions
  - Reduced energy per packet (data aggregation)



Key Issues in Data Aggregation

Main questions to be addressed:

- which routing topology to use?
- how does a node merge multiple packets into a single one?
- when does a node report a sensed event?

Heuristics

- Center at Nearest Source (CNSDC): All sources send the information first to the source nearest to the sink
- Shortest Path Tree (SPTDC): merge the shortest paths from each source wherever they overlap
- Greedy Incremental Tree (GITDC): Start with path from sink to nearest source; add next nearest source to the existing tree

Data Aggregation Issues

Many problems depending on chosen parameters

- centralized vs distributed aggregation policies (distributed: data aggregation occurs locally at each node using local observations)
- time synchronous vs time asynchronous network
- reporting
  - periodical reporting
  - base station inquiry response reports for sensed information
  - event triggered reports: the occurrence of an event might trigger reports from sensors in that region

Data Aggregation Issues (2)

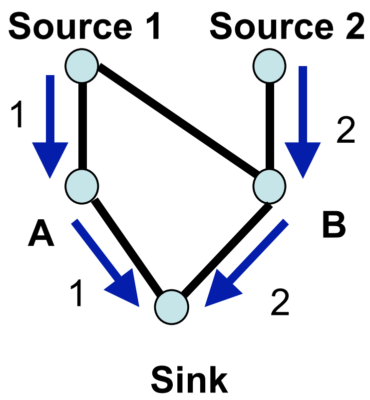
- objective function
  - min total energy cost
  - max network's lifetime
- aggregation function: energy requirement for transmitting aggregated packets
  - given by the specific encoding
  - concave function of packet size
  - aggregation savings depend on spatial information

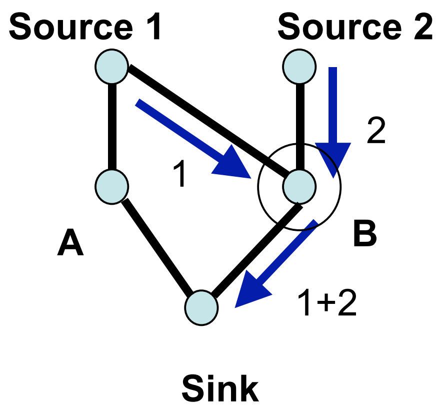
Data Aggregation Issues (3)

- routing network: hierarchical, tree, clusters of sensors, dynamically modification of routing

Other issues..

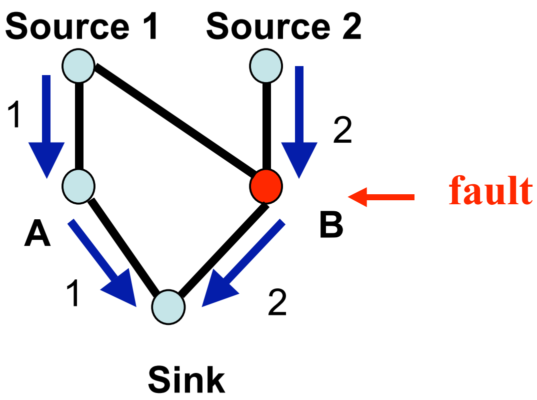
- fault tolerance

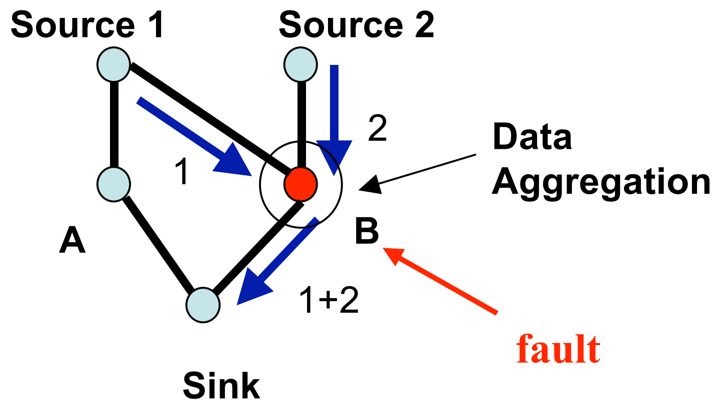




Data Aggregation Issues (3)

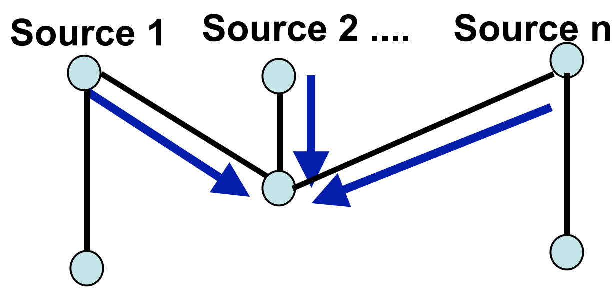
- fault tolerance





Data Aggregation Issues (4)

- interferences during transmissions



- relationships with other techniques for energy saving (eg wake-asleep cycles)
- ....

A known special case

given

- a graph G with 1 sink node
- a subset of m nodes that should report to the sink (periodic report)
- fully synchronized network
- transmission costs among nodes (per bit)
- an aggregation function specifies compression of packets (independent of spatial location)

find a routing tree that minimize the total transmission cost

Single source buy at bulk

given

- a graph G with 1 source node
- a subset S of m nodes that should receive data from the source

goal

- cost of an edge: concave function of # bit sent through the edge
- buy edges to obtain a tree connecting the source with all nodes in S

- Steiner tree as a special case
- approximation results
  - Goel Estrin 03: algorithm for any concave function
  - constant approximation algorithm ...

Problem 1

previously used objective function:

min total energy cost

- finding routing tree of minimum energy cost
  overloaded sensors might run out of energy

Maximum Lifetime Data Aggregation

goal: maximize network's lifetime

- share the transmission cost among sensors
- dynamically modification of routing
  - routing tree is changed based on sensor energy
  - clusters of sensors

Problem 2

previous solution (buy at bulk) is based on time synchronous networks and periodic reporting

in general, given a routing tree

- nodes should wait for a certain period of time before they fuse the received reports
- a sensor node may timeout before receiving reports from all of its children

energy-latency tradeoff

- with insufficient reports, the credibility of a sensed event is questionable
- waiting too much causes late reports (that might be useless)

Problem 1

Maximum Lifetime Data Aggregation (MLDA)

Maximum Lifetime Data Aggregation (MLDA)

The MLDA problem is to find a data gathering schedule with maximum lifetime [Dasgupta,Kalpakis,Namjoshi 2003]

Given:

- the location & energy of each sensor and of the sink
- assume that at each time unit a packet is generated at each sensor (periodic reporting)

Find an efficient manner to collect & aggregate all reports from the sensors to the sink

- a feasible schedule is a schedule which respects the energy constraints of the sensors
- an optimal schedule maximizes T, network lifetime

MLDA: System Model

- n sensor nodes (1..n) one base station (n+1);
- locations of nodes are fixed and known
- continuous data delivery (round= 1 time unit)
  - at each round a sensor produces a packet of k bits
  - aggregation of packets of k bits gives one packet of k bits
- each sensor can transmit to any other sensor or to the base station
  - Initial energy of a sensor i: E_i
  - Receive energy, $RX_i = e_{elec} * k$
  - Transmission energy, from i to j

$$TX_{i,j} = e_{elec} *k + e_{amp} *d^2_{i,j} *k$$

MLDA Problem: solution

Algorithm

1. flow formulation with linear objective function and and integrality constraints on flow (O(n^3) variables)
2. LP is employed to find a near-optimal integral admissible flow network
3. A schedule is generated from the admissible flow network

- *very high time complexity*
- *Experiments show that this solution is good*

MLDA Problem: flow formulation

- given feasible values of

  $f(i,j)$ = no. packets sent from i to j
- G is the graph with node set given by sensors nodes and arc capacity given by $f(i,j)$ (for all i and j)
- $g(i,j,k)$ is the flow sent by node k through arc (i,j)

max T (T= network lifetime) s.t.

1. [T flows reaches the BS] $\sum_j g(j,n+1,k) = T$ for all k
2. $0 \le g(i,j,k) \le f(i,j)$ for all i,j,k
3. flow conservation constraints at each node
4. integrality constraints on flow variables $g(i,j,k)$

MLDA Problem: ILP formulation

f(i,j) no. of packets sent from i to j

g(i,j,k) flow sent by node k through arc (i,j)

max T s.t. (T= network lifetime)

1. [T flows reaches the BS] $\sum_j g(j,n+1,k) = T$ for all k
2. $0 \leq g(i,j,k) \leq f(i,j)$ for all i,j,k
3. flow conservation constraints at each node
4. [energy constraint at i]
   $\sum_j f(i,j)\, TX_{i,j} + \sum_j f(j,i)\, RX_i \leq Ei$ for all i
5. integrality constraints

MLDA Problem: ILP formulation

Given the ILP formulation

- LP is employed to find a fractional optimal admissible solution
- a flow formulation is obtained by rounding f(i,j) values
- an integer solution to the ILP formulation is computed by recomputing the solution with the floored f(i,j) as constraints

then

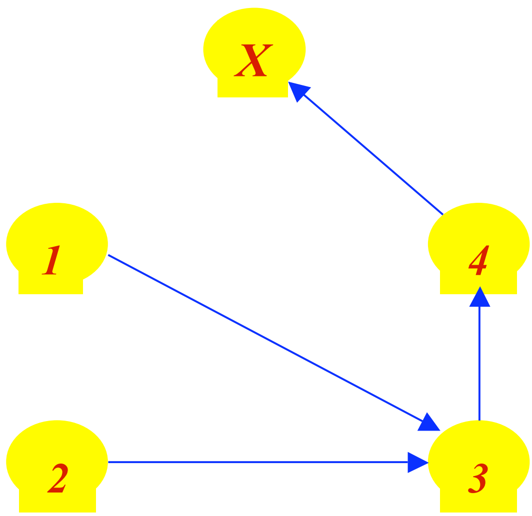
Given an integral solution with lifetime T

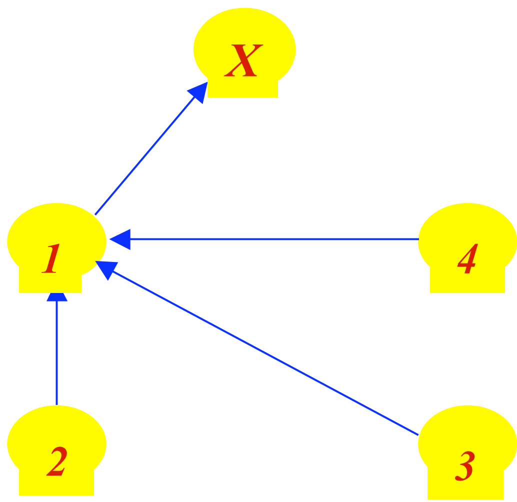
- determine the routing used for each round which allows in-network data aggregation

Maximum Lifetime Data Aggregation (MLDA) Problem

An aggregation tree is a directed tree rooted at the base station and spanning all the sensors

- it specifies how data packets are collected, aggregated and transmitted to base station.
- at each round the aggregation tree might change





Maximum Lifetime Data Aggregation (MLDA) Problem

- A schedule with lifetime T is a collection of up to T aggregation trees
- Given an integrality solution to the flow problem it is easy to get a set of aggregation trees, one for each round (using branching theory)

Further results

- Heuristics (K.Kalpakis, et al.)
  - G-CMLDA
  - I-CMLDA
- Garg-Könemann approx. alg. with minimum length columns instead of solving the linear programming
- other algortihms: minimum cost spanning arborescence problem

Problem 2

Latency Constrained Aggregation

Problem 2

Given a data fusion architecture (a routing tree)

- nodes should wait for a certain period of time before they fuse the received reports
- a sensor node may timeout before receiving reports from all of its children

Tradeoff

- with insufficient reports, the credibility of a sensed event is questionable
- waiting too much causes late reports (that might be useless)

Latency constrained aggregation

Energy-Latency tradeoff:

Aggregation of packets reduces energy consumption

- Drawback
  - Need to wait for possible packets to aggregate

    -> latency increase
- Possible objectives
  - Minimize *f* (latency, transmission cost)
  - Minimize transmission cost

    subject to bound on the latency

    [Becchetti, Korteweg, AMS, Stougie, Skutella, Vitaletti, 2006]

The Model

- Routing intree $T=(V, A)$ is given
  - root(T) is the sink, every node $v \in V$ is a sensor
  - arcs represent communication links:
    - $c(a)$: communication cost of arc a (energy)
    - $\tau(a)$: transit time of arc a
    - this talk: $c(a)$ and $\tau(a)$ are independent of the size of the packet
- A sensing event generates a message $j = (v_j, r_j, d_j)$
  - v_j release node
  - r_j release time
  - d_j due date

The Model cont.

Aggregation:

- 2 or more messages aggregated **->** messages are simultaneously sent in a packet
- recursive aggregation possible
- due date of packet equals earliest due date of a message in the packet

Problem: send all messages to the sink

- minimize total transit cost
- obey all due dates

The Model cont.

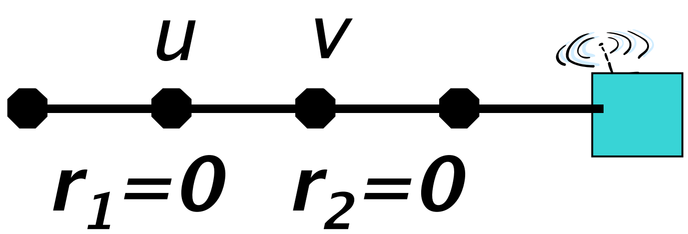
- Delayed transmission of a message/packet might favour aggregation

- Arrival interval of message $j = (v_j, r_j, d_j)$ is $[r'_j, d_j]$ where

  $r'_j := r_j +$ (total transit time from v_j to sink)
  - r'_j : earliest time j might reach the sink
  - $d_j - r'_j$ bounds the total waiting time of j

Example

- 2 messages: *M1= (u,0,4)* and *M2=(v,0,4)*
- for all arcs *a*: *□(a)=1, c(a)=1*
- if *M2* is immediately sent, then total cost is *5*
- if *M1* is immediately sent and *M2* waits, then *M1* and *M2* are aggregated and total cost is *3*



Optimal Solutions

There exists an optimal solution such that

1. if two messages are aggregated in a packet, they stay together until they reach the sink
2. a message waits only at its release node
3. a packet arrives at the sink at the earliest due date of any message in the packet

Clique Partitioning

- given a tree and a set of messages *M*, construct a graph *G =(M,E)* such that
  - a vertex of *G* correspond to a message in *M*
  - two vertices are adjacent iff the arrival intervals of the corresponding messages intersect
- clique in *G* is a set of messages that might be aggregated
- graph *G* is an interval graph

problem: find a clique partitioning of graph *G*

that minimizes a suitable objective function

Off-line Problem

Off-line problem has mainly theoretical interest:

- chain networks: polynomial
- intree: NP-complete (even for depth 2 tree or uniform costs)
- intree: 2-approximation (LP based)

ILP Formulation

binary variable x_{ia} is 1 iff arc a is used by some message j arriving at the sink at time d_i

a_j is the arc leaving the release node of message j

$$\min \sum_a c(a) \sum_i x_{ia}$$

$$\text{s.t.} \quad \sum_{i=j_{\min}}^{j} x_{ia_j} \geqslant 1 \; \forall \; j$$

$$x_{ia} \geqslant x_{ia'} \qquad \forall \; i \quad \forall \; a, a' \text{ with } \mathsf{head}(a') = \mathsf{tail}(a)$$

$$x_{ia} \in \{0, 1\}$$

LP Rounding

Lemma: Let $\alpha_1, \ldots, \alpha_n \in \mathbb{R}_{\geqslant 0}$ and $\beta_1, \ldots, \beta_n \in \{0, 1\}$ with

$$\sum_{i=j}^{k} \alpha_i \geqslant 1 \quad \Longrightarrow \quad \sum_{i=j}^{k} \beta_i \geqslant 1 \quad \forall 1 \leq j \leq k \leq n. \qquad (1)$$

By decreasing some of the β_i's from 1 to 0, one can enforce the inequality

$$\sum_{i=1}^{n} \beta_i \leqslant 2 \sum_{i=1}^{n} \alpha_i$$

while maintaining property (1).

On-line Problem

- Centralized Model
  - each node has full knowledge of the network topology and of messages in the network
- Synchronous Distributed Model
  - each node knows its distance from the sink and there is a common clock
- Asynchronous Distributed Model
  - each node knows only its distance from the sink

WR Algorithm

- WR means that messages can only *wait* at their *release node*;
- when a message reaches a node, *aggregation* is performed whenever possible.

WR algorithms are good in the synchronous but bad in the asynchronous model!

Synchronous Distributed Model

Algorithm Common Clock (CC):

a message j waits at its release node; it is sent to arrive at the sink at time $t(r'_j, d_j)$

For message j let

$$i := \max \{ i \mid \text{exists } k \text{ s.t. } k2^i \in [r'_j, d_j] \} ;$$

then k s.t. $k2^i \in [r'_j, d_j]$ is unique and we set

$$t(r'_j, d_j) := k2^i$$

Synchronous Distributed Model

Theorem: Algorithm CC achieves competitive ratio $O(\log U)$ where U is the ratio between the maximum and the minimum arrival interval length

Theorem: Any deterministic synchronous algorithm is $\Omega(\log U)$ competitive

WR for Asynchronous Distributed Model

Theorem: In the asynchronous distributed model

- any deterministic WR algorithm is □(m) competitive on a chain with m edges;
- any randomized WR algorithm is □(m) competitive on an intree with m edges.

Where do messages wait? How long?

Our proposal

Spread equally

- every message spends its waiting time equally at all nodes on its path to the sink
- aggregation is performed whenever possible
- Multi-level Fusion Synchronization (MFS) Protocol (Yuan 2003)

Theorem: Spread equally is O(□ log *U*) competitive where □ is the depth of the tree and *U* is the ratio between max and min arrival interval length.

Theorem: A more sophisticated algorithm achieves competitive ratio O[log□ (log□+log*U*)] for the case of a chain.

Our proposal

Further results

- **Theorem:** A more sophisticated algorithm achieves competitive ratio $O[\log \square (\log \square + \log U)]$ for the case of a chain
- All results hold if a concave aggregation cost function is udes instead of total aggregation

Recent results: extensions to the almost synchronous model (clocks have a small drift) using multicriteria objective function [*min f(energy,latency)*] [Korteweg,AMS,Stougie,Vitaletti 2007]

Conclusions

- Data aggregation can result in significant energy savings for a wide range of operational scenarios
- Two important aspects:
  - network's lifetime
  - synchronization among sensors
- Integration of these and other issues
  - spatial localization, distributed algorithms, interferences,...

give new interesting combinatorial optimization problems

Thank you!