An Energy-Efficient Strategy for Target Tracking through Wireless Sensor Networks

Loredana Arienzo, Maurizio Longo
Department of Electrical and Information Engineering
University of Salerno
Via P.te Don Melillo, I-84084 Fisciano (SA), Italy
Email: {larienzo, longo}@unisa.it

Abstract—In this paper, the problem of tracking cooperative mobile nodes in wireless sensor networks is addressed. Aiming at an efficient resource solution, the research adopts a strategy of combining target tracking with node selection procedures in order to select informative sensors to minimize the energy consumption of the tracking task. We layout a cluster-based architecture to address the limitations in computational, battery power and communications of the sensor devices. To track mobile nodes two particle filters, bootstrap and unscented particle filter, which estimate the actual position and predict future locations, are used. The particle filters have been widely used in tracking algorithms, but their energy efficiency has received less attention. To save energy, a node selection procedure is proposed. The node selection problem is formulated as a cross-layer optimization problem and it is solved using greedy algorithms.

Index Terms—Sensor networks, tracking, particle filters, energy efficient, cross-layer optimization, cluster.

I. INTRODUCTION

THE issue of *tracking targets* in Wireless Sensor Networks (WSNs) [1], i.e. to monitor the roaming path of moving objects in the area of deployment, has received significant attention in recent years [13], [17], [20]. Tracking a target in sensor networks is quite challenging regardless of the energy consumption due to resource-constrained of the network devices. The *sensor management* is the process of dynamically retasking sensors in response to an evolving environment. The goal of sensor management is to choose actions for individual sensors dynamically so as to maximize overall network utility. In a tracking task the sensor management addresses the problem of choosing informative sensors needed to obtain information about the target state and therefore maximize the network lifetime for a given cost of communication and computation. The problem of selecting the best nodes for tracking (in the mean squared position error sense) a target in a distributed wireless sensor network has been investigated by Estrin since 1999 [6]. Recently, information-theoretic node selection approaches have been proposed [10], [11], [22]. The main idea behind these approaches is to optimize an utility function, representing the location accuracy, using entropy-based metrics.

We propose an *information-based approach* that focus on maximizing an utility function, representing the overall energy in a cluster, using energy-based metrics. We show that properly selecting nodes to collect measurements in a cluster head, we can save energy to maximize the sensor network lifetime. The node selection is formulated as a cross-layer optimization aiming to minimize the total energy consumption in the cluster. While is widely recognized the maturity of the research area for cross-layer MAC, routing and power control design, much investigation on design regarding the topology level remain to be done. Topology of wireless networks can changes due to the mobility of nodes, the sleep mode, and the battery depletion. We propose a cross-layer design joining application level, topology control and energy management.

The contributions of the papers are as follows: (1) we introduce energy-based metrics to evaluate the energy consumption of node selection algorithms; (2) we formulate the node selection problem as a cross-layer optimization problem and determine the optimal solution by greedy algorithm; (3) we compare the proposed node selection algorithms with the existing literature; (4) we implement a distributed tracking algorithm using particle filters methods. The remainder of this paper is organized as follows: in Section II we describe the existing work. Section III provides the preliminary information by describing the model of the overall system. In Section IV we formulate the energy efficient tracking problem introducing the energy-based metrics and in the Section V we describe the optimization problem and provide solutions. Section VI discusses the performances of proposed algorithms, while in Section VII, we draw the main conclusions.

II. RELATED WORK

Many criteria influence the design of energy-efficient tracking approaches, and a wide range of schemes have been proposed. This Section is devoted to provide a survey of existing target tracking techniques with the aim to finalize our study on energy-efficient approaches. Table I presents a classification of these approaches.

Generally, the *hierarchical structures* include tree-based, cluster-based and prediction-based structures. The tree-based approaches [12], [14], [24], [25] use a hierarchy tree to represent the sensors and record information about the presence of objects detected by the sensors. Kung *et al.*

GTTI'08, June 16–18, 2008, Florence, Italy.

TABLE I
Classification of Target Tracking according to different criteria.

Criteria	Classification
Hierarchical structures	Tree-based vs. cluster-based
Topology information	Global vs. local
Signal processing	Centralized vs. distributed
Sensor management	Information-driven vs.information-based

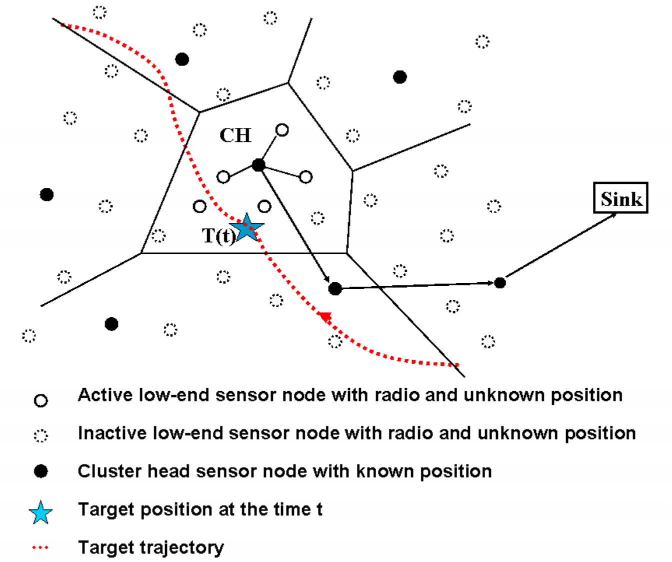


Fig. 1. Sensor Network Topology.

[12] propose STUN (Scalable Tracking Using Networked Sensors), a scalable tracking architecture that employs hierarchical structure to allow the system to handle a large number of tracked objects. Additionally to the tree, Lin *et al.* [14] consider an in-network moving object tracking in a sensor network, consisting of two operations: location update and query. The drawback is the building of the tree as the target moves. Zhang *et al.* [24], [25] propose DCTC (Dynamic Convoy Tree-Based Collaboration). They introduce a message-pruning tree structure called convoy tree, which is dynamically configured to add and prune some nodes as the target moves and the tracking problem is formalized as a multiple objective optimization problem. The solution to the problem is a convoy tree sequence with high tree coverage and low energy consumption. Building such a convoy tree sequence requires global network information, and re-configuration and maintenance of a convoy tree incurs considerable computational and communication overhead. As a result, the tree-based approaches are usually centralized and applied in the deployment phase of sensor networks.

Wang *et al.* [23], Chen *et al.* [5], propose a cluster-based tracking scheme. They envision a hierarchical sensor network that is composed of (a) a static backbone of sparsely placed high-capability sensors which will assume the role of a cluster head (CH) upon triggered by certain signal events; and (b) moderately to densely populated low-end sensors whose function is to provide sensor information to CHs upon request. In these schemes, sensors are grouped into clusters either statically or dynamically (upon detection of the target in the vicinity), and a cluster head collects information from its cluster members and determines the target location using either the trilateration technique [23] or the Voronoi diagram-based approach [5]. Both localization approaches aim to determine the exact location of the target at the expense of considerable computational overhead.

From the *topology* perspective, the tracking approaches could use a global or local knowledge about the location of every node in the network. As opposed to the tree-based schemes [12], [14], [24], [25] that use a global information, the cluster-based schemes [5], [23] relies on local topology knowledge to limit the scope of target's location updates.

Based on the *sensor management* or *collaborative data processing*, the existing approach of target tracking can be classified in information-driven and information-based. Zhao *et al.* [26] propose IDSQ (Information Driven Sensor Querying), in which the selection of the best node is based on a Mahalanobis distance that leads to a heuristic method favoring the sensors whose Euclidean distance to the target is small. In [10], [11], [22] the focus is on information-based approaches, i.e. the heuristics select an informative sensor such that the fusion of the selected sensor observation with the prior target location distribution would yield to minimize the entropy of the target location distribution.

As to the *signal processing*, the tracking approaches can be classified as centralized or distributed. Usually the tree-based are centralized approaches, while the cluster-based are distributed schemes in which the cluster head is the leader node in the processing. Reference [15] is a centralized approach, while [9], [13], [20], [26] are distributed approaches. Classical tracking [4] is often formulated as a Kalman filtering problem, using Gaussian noise distributions for the sensor measurements and the target trajectory. Distributed tracking based on Kalman filtering has recently been considered [10], [11]. *Particle filters* [3], [8] offer an alternative to Kalman filtering in nonlinear non-Gaussian problems, and have been investigated for tracking applications in sensor networks [17], [20]. While Kaplan [10], [11] estimates the target location using a Kalman filter based on the current measurement at a sensor and the past history at other sensors, we propose a distributed tracking based on particle filters.

III. SYSTEM MODEL

We make three assumptions about the sensor network. First, the network is composed by a single gateway (sink) node and multiple sources (Fig. 1). Next, the network is modeled as a combination of 1) a static backbone of sensors which assume the role of a cluster head and 2) randomly distributed low-end sensors which sense a moving target and report data to CHs upon request. Finally, we assume that the network is composed of dynamic clusters, depending on the predicted target trajectory. According to the dynamic clustering [9], a cluster is formed and a CH becomes active in an on-demand fashion, when the received signal strength (RSS) of the CH exceeds a predetermined threshold. A local topology knowledge by the construction of Voronoi diagrams at the CH is used to limit the scope of target-initiated location updates. The sensor of the cluster, which will assume the role of cluster head, will predict the trajectory of a target by means of a particle filter based on the history of the target location and some observations which come from some active sensors in the cluster. These sensors will be referred as *active sensors* (Fig. 1). The details of the clustering algorithm are out of the aim of this paper. In the following we will limit ourselves to consider only the inter-cluster communication issues.

A. ENERGY MODEL

The main concept underlying the proposed energy efficient tracking algorithm is that the CH should select the active neighbors with the goal of reducing the total energy consumption in the cluster. In this Subsection we introduce two metrics based on energy consumption for a tracking task.

1) Energy-based Metric: To describe the energy consumption, we use the energy model for wireless sensor networks introduced by Wang *et al.* [21]. The energy consumption per bit at the physical layer is:

$$E = E_{Tx-elec} + \beta d^{\alpha} + E_{Rx-elec} \tag{1}$$

where $E_{Tx-elec}$ and $E_{Rx-elec}$ are distance independent terms that takes into account overheads of transmitter electronics (and digital processing) and receiver electronics, respectively; finally βd^{α} accounts for the radiated power necessary to transmit one bit over a distance d, where α is the exponent of the path loss ($2 \leq \alpha \leq 5$). According to [9] we assume that $E_{Tx-elec} = E_{Rx-elec} = E_{elec}$. Hence, given l bits of data, the overall energy consumption to transmit the packet of l bits between two nodes at a distance d with a given received SNR can be expressed as:

$$E(d,l) = (2E_{elec} + E_{amp} \cdot d^{\alpha}) \cdot l \tag{2}$$

where E_{elec} $[Joule/bit]$ is the energy needed by the transceiver circuitry to transmit or receive one bit and E_{amp} $[Joule/(bit \cdot m^{\alpha})]$ is a constant which represents the energy needed to transmit one bit over a distance d to achieve an acceptable SNR at the destination. This model assumes that the energy consumption is dominated by the radio communication rather than the computation. We refer to (2) as the energy-based metric.

2) Combined Energy-based Metric: According to [18] we introduce another metric combining energy consumption and remaining energy at nodes. Hence, if we refer to a link (i,j) with distance d, the overall energy consumption to transmit a l-bit packet between the node i and node j with a given received SNR can be expressed as:

$$E_{ij}(d,l) = E_r - (2E_{elec} + E_{amp} \cdot d^{\alpha}) \cdot l \tag{3}$$

where the first addend E_r is the remaining energy at node i and the second addend is the transmission energy required for the node i to transmit l bits to its neighboring node j. We refer to (3) as the combined energy-based metric.

B. NETWORK MODEL

In our analysis, we consider a network composed by sensor nodes deployed according to a 2-D Poisson distribution. We assume to know the position of the CH (static node) and to estimate the distance of each neighbor with respect to the CH using the power of a radio signal emitted by the object. Therefore, we assume the log-normal shadowing model for the channel and we suppose that the power (in dB) of a received signal decreases exponentially with the propagation distance:

$$P_r(d) = P_r(d_o) \cdot (d_o/d)^{\alpha} + X_{\sigma} \tag{4}$$

where $P_r(d)$ is the received power at a receiver at distance d from a transmitter, $P_r(d_o)$ is the transmitted power at a reference distance d_o, α is the path loss exponent, and X_{σ} is the shadow fading component, with X_{σ} Gaussian distribution $\mathcal{N}(0, \sigma)$. Hence, the distance from the i-th sensor of the cluster to the CH can be estimated as $d = (P_r/P_a)^{-1/\alpha}$, where P_r is the RSS at the sensor, P_a is the (unknown) strength of the signal from the sensor.

IV. JOINT TRACKING, TOPOLOGY CONTROL AND ENERGY MANAGEMENT

The aim of this section is to describe the proposed cross-layer tracking algorithm.

A. TRACKING USING PARTICLE FILTERING

We use sequential MC (SMC) approaches, also known as particle filtering, for tracking a moving target. The particle filter provides simulation-based solutions to estimate the posterior distribution of nonlinear discrete time dynamic models. The required posterior distribution density of the target is represented by a set of random samples with associated weights. To estimate the target's location these samples (or particles) and the associated weight are recursively updated using the sequential importance resampling (SIR) algorithm. The particle filter using SIR techniques is known as *bootstrap filter* or SIR particle filter (PF) [8]. A common problem with SIR is the degeneracy phenomenon, i.e. the samples may eventually collapse to a single point if, during the resampling stage, samples with high importance weights are duplicated an extremely large number of times. Using the prior distribution as importance distribution, as in PF, could lead to the degeneracy problem of the particles because of the most recent observations are ignored. There have been numerous proposals to rectify the degeneracy problem improving the performance of the SIR particle filter [3]. Notable techniques include local linearization using the extended

Kalman filter (EKF) [4] or the unscented Kalman filter (UKF) to estimate the importance distribution [19]. A particle filter which uses UKF to generate the importance distribution is referred as *unscented particle filter* (UPF) or sigma-point particle filter [20]. We have implemented the bootstrap particle filter and the unscented particle filter and we have compared their performances.

In this paper we assume as dynamic model of the target the constant velocity model [20]. Hence, denoting by $\mathrm{x}_k = [\alpha_k, \dot{\alpha_k}, \beta_k, \dot{\beta_k}]^T$ the state vector (coordinates along x, y axes and the velocities) of a target, the state-space model is given by,

$$\mathrm{x}_{k+1} = \begin{pmatrix} 1 & \Delta\mathrm{T} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta\mathrm{T} \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathrm{x}_k + \begin{pmatrix} \Delta\mathrm{T}^2/2 & 0 \\ \Delta\mathrm{T} & 0 \\ 0 & \Delta\mathrm{T}^2/2 \\ 0 & \Delta\mathrm{T} \end{pmatrix} \mathrm{v}_k$$

where $\mathrm{v}_k \sim \mathcal{N}(0, diag(\sigma_x^2, \sigma_y^2))$ denotes the motion noise and ΔT the length of a snapshot.

Additionally, as observation model of the measurements, we use the log-normal shadowing model. Hence, let $\{\alpha^s, \beta^s\}$ be the fixed position of sensor s and $d_k = [(\alpha_k - \alpha^s)^2 + (\beta_k - \beta^s)^2]^{(1/2)}$ be the distance between the sensor s and the target, in a logarithmic scale the target-originated measurement are modeled by

$$\begin{aligned} \mathrm{h}_k(x_k) &= K - 10\alpha \log(d_k) \\ \mathrm{y}_k &= \mathrm{h}_k(x_k) + \mathrm{n}_k \end{aligned} \tag{5}$$

where the measurement noise n_k accounts for the shadowing effects and other uncertainties. The noise n_k, as in Subsection III-B, is assumed to be a zero-mean Gaussian, i.e. $\mathrm{n}_k \sim \mathcal{N}(0, \sigma)$, and the sensor noises are assumed uncorrelated; K is the transmission power, and α is the path loss exponent.

B. ENERGY EFFICIENT TRACKING

We denote with $\mathcal{N}_a$ the neighbor nodes' set of the cluster with RSS exceeding a predetermined threshold and with $\mathcal{N}_d$ the desired anchor nodes' subset needed for the tracking algorithm. According to the metric (2) and the location discovery protocol in [2], the energy cost in the communication with the sensor $i \in \mathcal{N}_a$ is given by:

$$\begin{aligned} E_i(r_i, d_i) &= [E_{elec}(N_d + 2) + E_{amp} \cdot d_i^{\alpha}] \cdot \frac{l}{T_M} \\ &+ [E_{elec}(N_d + 1) + E_{amp} \cdot r_i^{\alpha}] \cdot b \end{aligned} \tag{6}$$

where b is the bit rate [bit/s] between the CH and the node i, T_M [s] is the period between two consecutive discovery signal of the target, r_i and d_i are, respectively, the distance of the node i from the CH and the target, while N_d is the number of desired nodes for the tracking task. Finally, in (6) $E_{elec}N_d$ represents the energy needed at the neighbors to receive one bit. In the energy cost we have omitted the energy consumption in the path between the target and the CH due to the calibration phase of the clustering and we have considered only the communication between each node of the cluster and its cluster head.

Algorithm 1 Greedy Random Node Selection

```
Synopsis: $[\mathcal{N}_d, \mathcal{C}, E_{tot}, node, E_b, k]$ =Greedy($\mathcal{N}_a$,$N_d$).
Given: Set of nodes in the cluster $\mathcal{N}_a$, number of desired nodes $N_d$.
Output: Set of desired nodes $\mathcal{N}_d$, new set of candidate nodes in the cluster $\mathcal{C}$, total energy of the desired set $E_{tot}$, last node selected in the current snapshot $node$, energy of this node $E_b$, time $k$.

Initialize the candidate set:
    $\mathcal{C} = \mathcal{N}_a$
    $\mathcal{N}_{cand} = 0$
Initialize the objective function:
    $E_{tot} = 0$
Randomly select a candidate node $i \in \mathcal{C}$
    $E_{min} = E(i)$
    $NodeMin = i$
while $|\mathcal{N}_{cand}| < N_d$ do
    for each $j \in \mathcal{C} \setminus \{i\}$ do
        if $E(j) < E_{min}$ then
            $E_{min} = E(j)$
            $NodeMin = j$
        end if
    end for
    $node = NodeMin$
    $E_b = E_{min}$
    $E_{tot} = E_{tot} + E_{min}$
    $\mathcal{N}_{cand} = \mathcal{N}_{cand} \cup \{NodeMin\}$
    $\mathcal{C} = \mathcal{C} \setminus \{NodeMin\}$
end while
$\mathcal{N}_d = \mathcal{N}_{cand}$
```

Therefore, the total energy cost, for all nodes in the set $\mathcal{N}_a$, is given by

$$E^{TOT}(\mathcal{N}_a) = \sum_{i \in \mathcal{N}_a} E_i(r_i, d_i) \tag{7}$$

V. PROBLEM FORMULATION

Our objective is to select the optimal subset $\mathcal{N}_d \subset \mathcal{N}_a$ which minimizes the total energy cost 7, i.e.

$$\mathcal{N}_d = arg \min_{\mathcal{N} \subseteq \mathcal{N}_a} E^{TOT}(\mathcal{N}) \tag{8}$$

The solution of this optimization problem is illustrated in the following Subsections.

A. THE SOLUTION IN THE STATIC SCENARIO

To find the optimal solution for such problem it is theoretically possible to enumerate the solutions and evaluate each with respect to the stated objective. However, from a practical perspective, it is infeasible to follow such a strategy because the number of combinations grows exponentially with the size of problem. If we formulate our combinatorial optimization problem as an ILP (Integer Linear Programming) problem, the computational complexity consists of enumerating all the N_d-node subsets, $\mathrm{O}(N_a^{N_d})$, and adding the computational complexity of the assignment problem, $\mathrm{O}(N_d^3)$. In such cases, heuristic methods are usually employed to find good, but not necessarily guaranteed optimal solutions. More than one technique is applicable, i.e. integer linear programming, graph theory, Algorithm 2 Branch and Bound Algorithm

```
Synopsis: $[\mathcal{N}_d, \mathcal{C}', E_{tot}] = BranchBound(N_d, \mathcal{C}, E_b, node)$
Given: Number of desired nodes $N_d$, candidate nodes set of the cluster $\mathcal{C}$, energy bound $E_b$, node related to the energy bound $node$.
Output: Set of desired nodes $\mathcal{N}_d$, new set of candidate nodes in the cluster $\mathcal{C}'$, total energy of the desired set $E_{tot}$.

Initialize the candidate set:
    $\mathcal{N}_{cand} = \{node\}$
Initialize the objective function:
    $E_{tot} = E_b$
while $|\mathcal{N}_{cand}| < N_d$ do
    for each $j \in \mathcal{C} \setminus \{i\}$ do
        if $E(j) < E_b$ then
            break
        else
            $E_{min} = E(j)$
            $NodeMin = j$
        end if
    end for
    $\mathcal{N}_{cand} = \mathcal{N}_{cand} \cup \{NodeMin\}$
    $\mathcal{C} = \mathcal{C} \setminus \{NodeMin\}$
    $E_{tot} = E_{tot} + E_{min}$
end while
$\mathcal{N}_d = \mathcal{N}_{cand}$
```

genetic algorithms, greedy heuristics, see [16] for further details. We adopt the meta-heuristic GRASP (Greedy Randomized Adaptive Search Procedure) [7], in which each iteration consists of two phases, a construction phase, in which a feasible solution is produced, and a local search phase, in which a local optimum in the neighborhood of the constructed solution is sought. The best overall solution is kept as the result.

The implementation of the optimal Greedy node selection procedure is described in Algorithm 1.

B. THE SOLUTION IN THE DYNAMIC SCENARIO

In previous sections, we have considered the static version of the problem, namely a snapshot model. In this section we extend the Greedy node selection procedure over multiple snapshots, so that we can select active nodes for the next measurement intervals. In a dynamic scenario, due to the target mobility, the distance d_i in the Eq. (6) varies with the time and hence the total energy cost (7) is a function of time k.

In the dynamic version of the optimization problem we use the *dynamic programming* [16], that is based on the idea of breaking down the problem into stages at which the decisions take place and finding a recurrence relation that takes us backward from one stage to the previous stage. For this purpose, a branch-and-bound method is developed, in which the branch refers to the partitioning process into stages, that are repeatedly decomposed until a solution is found or infeasibility is proved, and the bound refers to lower bounds that are used to construct a proof of optimality without exhaustive search. We introduce an *energy bound* as in the following definition.

Algorithm 3 Tracking Algorithm

```
Synopsis: $[\mathcal{N}_d, \mathcal{C}, E_{tot}]$ =DynamicSelection($\mathcal{N}_a$,$N_d$).
Output: Set of desired nodes $\mathcal{N}_d$, new set of candidate nodes in the cluster $\mathcal{C}$,total energy of the desired set $E_{tot}$.

$[\mathcal{N}_d, \mathcal{C}, E_{tot}, node, E_b, k]$ =Greedy($\mathcal{N}_a$,$N_d$)
Loop until time runs out:
    Particle filtering to estimate the target's trajectory.
    Update the candidate set $\mathcal{C}$ during the dynamic of the target.
    $[\mathcal{N}_d, \mathcal{C}', E_{tot}]$ =BranchBound($N_d, \mathcal{C}, node, E_b$)
```

DEFINITION 1: *The* energy bound *is the maximum energy referred to the energy costs connected to the nodes which have been selected in the previous snapshot.*

Algorithms 2 provides pseudo-code of an efficient implementation of our branch-and-bound approach. Finally, in Algorithm 3 has been reported the overall tracking algorithm which combine the node's selection procedures with the particle filtering algorithm.

VI. PERFORMANCE EVALUATION

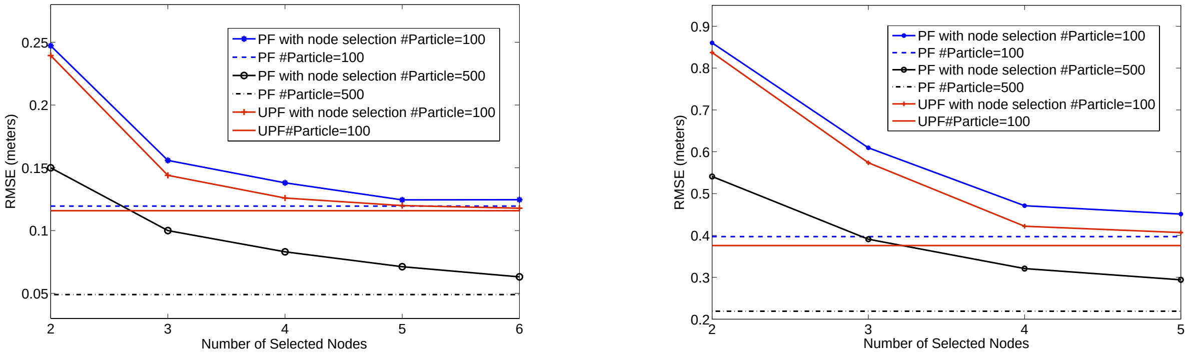
In this section, we investigate the performance of the overall target tracking system looking first at the node selection algorithm and then at the tracking algorithm.

A. OPTIMAL NODE SELECTION

In the following we compare our Greedy node selection algorithm with the *Kaplan* algorithm in [10] and [11]. The only difference between [10] and [11] is that in [10] the global topology knowledge is assumed, in which every active node reaches the entire network, while in [11] the only knowledge of the relative position to the target and the active nodes from the previous snapshot is required. For each iteration the computational complexity of Greedy algorithm is $O(N_a - N_d)$ while the computational complexity of Greedy algorithm is $O((N_a-2)^2)$, using N_d as number of desired node. Hence, the computational complexity for all iterations is given by

- Greedy Computational Complexity: $\sum_{i=0}^{N_d}(N_a - i) = N_a N_d + N_a - \frac{N_d^2}{2} - \frac{N_d}{2}$.
- Kaplan Computational Complexity: $\sum_{i=0}^{N_d}(N_a - i)^2 = N_a^2(N_d+1) + \frac{N_d(N_d+1)(2N_d+1)}{6} - 2N_a\frac{N_d(N_d+1)}{2}$.

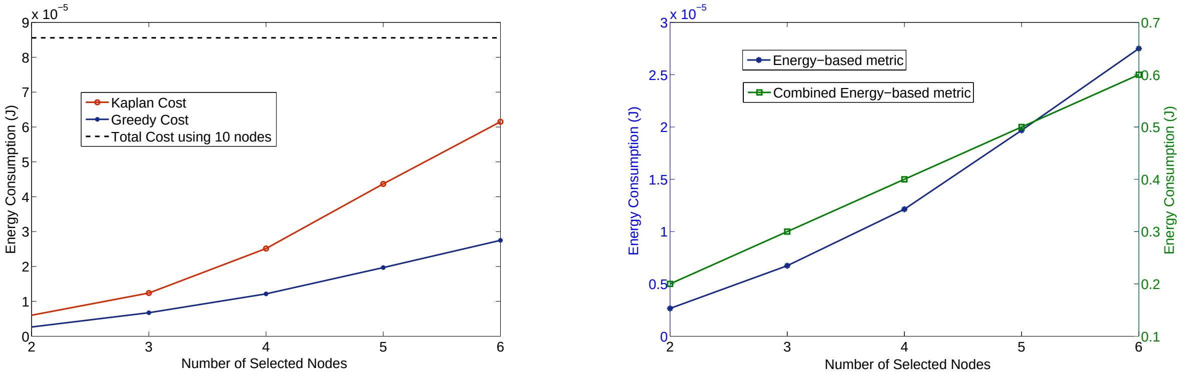
Indeed, in the above analysis of the *Kaplan* algorithm we have omitted the computational complexity of the initialization step of the Simplex algorithm, in which two nodes are chosen by exhaustive search. Definitely, due to computational complexity and because the simplex does not always find the global minimum, our approach outperforms the *Kaplan* algorithm.



(a) Scenario 1: target velocity=0.1 m/s

(b) Scenario 2: target velocity=0.3 m/s

Fig. 2. Performance accuracy comparison of particle filters. (K=9 dB, α=3, E_{elec}=10 nJ/bit, E_{amp}=100 pJ/bit/m^3, T_M=2 sec, b=10 bit/sec, l=8 bits, ΔT=1 sec, Observation Variance=0.3, E_r=100 mJ, Sensing Range=30 m.)



(a)

(b)

Fig. 3. Energy consumption of node selection algorithms using Na=10.

B. SIMULATION RESULTS

We implemented the node's selection algorithms and the particle filters in a Matlab simulator. We present simulation results for two scenarios: in Scenario 1 the nodes are randomly deployed on an area of 20m^2 and the target speed is 0.1 m/s; in Scenario 2 the nodes are randomly deployed on an area of 200m^2 and the target speed is 0.3 m/s. Fig. 2 shows the root-mean-squared error (RMSE) on the target's position of different filters versus the number of desired nodes using 100 runs. The performances of PF using two different number of particles are compared with the performance of UPF. Confidence intervals are not shown for the sake of clarity. The bounds of the errors are referred to the tracking algorithm without the node's selection procedure using a number of active nodes in the cluster equals to 10. Clearly, the filter with 500 particles outperforms the filter with 100 particle, i.e. the error decreases, but, as we will show the computation time increases.

In Fig. 2(a) a process variance σ_p equal to 1.0 has been used. The PF gives disappointing results with the low number of particle. As expected, the error should decreases as the number of nodes increases while the RMSE of PF, using a number of particles equal to 100, increases slightly with the number of desired nodes. We believe this is due to the drawbacks of the bootstrap particle filter, i.e. the degeneracy phenomenon and the sample impoverishment, that leads to a loss of diversity among particles; therefore particles will collapse to a single point within a few iterations. We point out that increasing the number of particles, using 500 particles as in Fig. 2 (a) the error of PF decreases as expected. To overcome these drawbacks in order to achieve smaller errors, we have investigated the effect of using the UPF.

Fig. 2(b) shows the RMSE for a process variance σ_p equal to 1.8. Again the UPF with 100 particles gives best results than the PF using 100 particles. Simulation results indicate a decrease in tracking performance with increase of noise and fast target movement. Note that, in Fig. 2(b) the values of the error when the number of desired nodes is equal two are omitted because in this case the filter overcomes. Other simulation results that we have not reported, with target velocity equal to 0.5 and 1.0 m/s, show that to estimate the track when the velocity increases a high number of anchor nodes is needed.

In Table II the results of a runtime measurement are illustrated, conducted on a system with AMD Opteron XP Processor 250, approx. 2400 MHz frequency and 4,00

TABLE II
Time to process greedy and Kaplan algorithms.

N_d	Greedy Time (s)	Kaplan Time (s)
2	6.6639e-5	13.9816e-4
3	7.8829e-5	32.5238e-4
4	1.0095e-4	10.0543e-3
5	1.2099e-4	23.1352e-3
6	1.4041e-4	52.2764e-3

GB RAM. Table II provides the execution time of the node's selection algorithms versus the number of desired nodes using N_a equal to 10. For each value of N_d, 10000 different random configurations were generated, where for each configuration, we assume a maximum range between node and target equal to 30 meters, and a maximum range between node and cluster head equal to 10 meters. The time to process PF with 100 and 500 particles is equal to 1,605 sec and 8,052 sec respectively, with three active nodes, while the time to process UPF with 100 particles is equal to 3,281 sec. In conclusion, UPF is less computational efficient than PF but performs a more accurate estimation of the target's position compared to PF.

Fig. 3(a) shows the energy consumption of the proposed greedy algorithm and Kaplan algorithm vs. the number of nodes N_d for a network size N_a equal to 10 nodes. Finally, in Fig. 3(b), we compare the energy consumption of the Greedy algorithm as the number of selected nodes increase, using the two different energy-based metrics defined in (2) and (3). Simulation results indicate an increase of the energy consumption with growing number of nodes. We highlight that the rise of the greedy algorithm energy consumption is superlinear using the energy-based metric while the rise is linear using the combined energy-based metric. Therefore, definitively can be concluded that the greedy selection algorithm outperforms the Kaplan selection algorithm to select the sensors that would give the most prolonged life to the network.

In conclusion, the energy consumption increases with the number of active nodes; on the other hand the tracking error decreases as the the number of active nodes increases. The proposed cross-layer approach shows that a tradeoff between the performance and the number of nodes is needed to save energy.

VII. CONCLUSION

We addressed the problem of saving energy in wireless sensor networks by introducing a node's selection procedure and assuming a cluster mechanism for the hierarchical structure. The node selection procedures were integrated into a particle filter and tested on simulated data. The experiments highlighted that the proposed approach outperforms the existing node's selection algorithms in literature. Extensive simulations showed that the target tracking system yields good accuracy for lower velocities of the target. The approach has been implemented with PF and UPF for nonlinear Gaussian problems. Finally, in this study we assumed an inter-cluster communication and limited ourselves to consider a single cluster in the evaluation of the energy consumption. Current work is investigating the implementation of algorithms to report tracking samples to multiple cluster heads.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, Wireless Sensor Networks: A Survey, *Computer Networks* (2002), 393–422.

[2] Loredana Arienzo, Efficient Particle Filter for Target Tracking in Wireless Sensor Networks, *ACM SIGCOMM*. Poster Session (Kyoto, Japan), August 27- 31 2007.

[3] M. S. Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp, A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, *IEEE Transactions on Signal Processing* 50 (2002), no. 2, 174–188.

[4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, Estimation with applications to tracking and navigation, 2001.

[5] Wei-Peng Chen, Jennifer C. Hou, and Lui Sha, Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks, *IEEE Trans. on Mobile Computing* 3 (2004), no. 3, 258–271.

[6] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, Next century challenges: Scalable coordination in sensor networks, *ACM/IEEE Mobile Computing and Networking* (Seattle, Washington), August 1999, pp. 263–270.

[7] T.A. Feo and M.G.C. Resende, Greedy randomized adaptive search procedures, *Int. J. of Global Optimization* 6 (1995), 109–133.

[8] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, Novel approach to nonlinear/nonlinear gaussian bayesian state estimation, *IEEE Proceedings – F* 140 (1993), no. 2, 107–113.

[9] W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan, An Application-Specific Protocol Architecture for Wireless Microsensor Networks, *IEEE Transactions on Wireless Communications* 1 (2002), no. 4, 660–670.

[10] Lance M. Kaplan, Global node selection for localization in a distributed sensor network, *IEEE Trans. on Aerospace and Electronics Systems* 42 (2006), no. 1, 113–135.

[11] ———, Local node selection for localization in a distributed sensor network, *IEEE Trans. on Aerospace and Electronics Systems* 42 (2006), no. 1, 136–146.

[12] H. T. Kung and D. Vlah, Efficient Location Tracking Using Sensor Networks, *WCNC*, March 2003.

[13] J. Lee, K. Cho, S. Lee, T. Kwon, and Y. Choi, Distributed and energy-efficient target localization and tracking in wireless sensor netwoks, *Computer Communic.* 29 (2006), 2494–2505.

[14] Chih-Yu Lin and Yu-Chee Tseng, Structures for In-Network Moving Object Tracking in Wireless Sensor Networks, *BROADNETS*, 2004.

[15] J. Liu, J. Reich, and F. Zhao, Collaborative in-network processing for target tracking, *EURASIP*, *J. Appl. Signal Processing*, 2003, pp. 378–391.

[16] C.H. Papadimitriou and K. Steiglitz, *combinatorial optimization*: *algorithms and complexity*, Prentice-Hall Inc., 1982.

[17] N. Shrivastava, R. Mudumbai, U. Madhow, and S. Suri, Target tracking with binary proximity sensors: fundamental limits, descriptions, and algorithms, *SenSys*, 2006, pp. 251–264.

[18] Ivan Stojmenovic and Xu Lin, Power-aware localized routing in wireless networks, *IEEE Trans. on Parallel and Distributed Systems* 12 (2001), 1122–1133.

[19] R. Van der Merwe, A. Doucet, N. de Freitas, and E. Wan, The Unscented Particle Filter, *NIPS*, 2000, pp. 584–590.

[20] T. Vercauteren and X. Wang, Decentralized Sigma-Point Information Filters for Target Tracking in Collaborative Sensor Networks, *IEEE Trans. on Signal Processing* 53 (2005), no. 8.

[21] A. Wang, W.B. Heinzelman, A. Sinha, and A. P. Chandrakasan, Energy-scalable protocols for battery-operated microsensor networks, *J. of VLSI Signal Processing* 29 (2001), 223–237.

[22] H. Wang, G. Pottie, K. Yao, and D. Estrin, Entropy-based Sensor Selection Heuristic for Targrt Localization, *IPSN* (Berkeley, California), April 26-27 2004.

[23] Qixin Wang, Wei-Peng Chen, Rong Zheng, Kihwal Lee, and Lui Sha, Acoustic Target Tracking Using Tiny Wireless Sensor Devices, *IPSN*, 2003.

[24] W. Zhang and G. Cao, DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks, *IEEE Trans. on Wireless Communications* (2004).

[25] Wensheng Zhang and Guohong Cao, Optimizing Tree Reconfiguration for Mobile Target Tracking in Sensor Networks, *IEEE INFOCOM Conference on Computer Communications*, 2004.

[26] F. Zhao, J. Shin, and J. Reich, Information-driven dynamic sensor collaboration for tracking applications, *IEEE Signal Processing Magazine* 19 (2002), no. 2, 61–72.