



**HAL**  
open science

## On Energy Efficiency in Collaborative Target Tracking in Wireless Sensor Network: A Review

Oualid Demigha, Walid Khaled Hidouci, Toufik Ahmed

► **To cite this version:**

Oualid Demigha, Walid Khaled Hidouci, Toufik Ahmed. On Energy Efficiency in Collaborative Target Tracking in Wireless Sensor Network: A Review. Communications Surveys and Tutorials, IEEE Communications Society, 2012, 15 (3), pp.1210 - 1222. 10.1109/SURV.2012.042512.00030 . hal-00957917

**HAL Id: hal-00957917**

**<https://hal.science/hal-00957917>**

Submitted on 11 Mar 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# On Energy Efficiency in Collaborative Target Tracking in Wireless Sensor Network: A Review

Oualid Demigha <sup>\*†‡</sup>, Walid-Khaled Hidouci <sup>†</sup>, Toufik Ahmed <sup>‡</sup>

<sup>\*</sup>Ecole Militaire Polytechnique

PO BOX 17, Bordj EL-Bahri, 16111 Algiers. ALGERIA

Email: odemigha@labri.fr

<sup>†</sup>Ecole Nationale Supérieure d'Informatique

Oued-Smar, Algiers. ALGERIA

Email: w\_hidouci@esi.dz

<sup>‡</sup>LaBRI, Université Bordeaux 1

351, Cours de la libération

33405 Talence Cedex. FRANCE

Email: tad@labri.fr

**Abstract**—Energy-efficiency in target tracking applications has been extensively studied in the literature of Wireless Sensor Networks (WSN). However, there is little work which has been done to survey and summarize this effort. In this paper, we address the lack of these studies by giving an up-to-date State-of-the-Art of the most important energy-efficient target tracking schemes. We propose a novel classification of schemes that are based on the interaction between the communication subsystem and the sensing subsystem on a single sensor node. We are interested in collaborative target tracking instead of single-node tracking. In fact, WSNs are often of a dense nature, and redundant data that can be received from multiple sensors help at improving tracking accuracy and reducing energy consumption by using limited sensing and communication ranges. We show that energy-efficiency in a collaborative WSN-based target tracking scheme can be achieved via two classes of methods: sensing-related methods and communication-related methods. We illustrate both of them with several examples. We show also that these two classes can be related to each other via a prediction algorithm to optimize communication and sensing operations. By self-organizing the WSN in trees and/or clusters, and selecting for activation the most appropriate nodes that handle the tracking task, the tracking algorithm can reduce the energy consumption at the communication and the sensing layers. Thereby, network parameters (sampling rate, wakeup period, cluster size, tree depth, etc.) are adapted to the dynamic of the target (position, velocity, direction, etc.). In addition to this general classification, we discuss also a special classification of some protocols that put specific assumptions on the target nature and/or use a “non-standard” hardware to do sensing. At the end, we conduct a theoretic comparison between all these schemes in terms of objectives and mechanisms. Finally, we give some recommendations that help at designing a WSN-based energy efficient target tracking scheme.

**Index Terms**—Energy conservation, prediction, state estimation, network self-organization, target tracking, collaborative signal processing, WSN.

## I. INTRODUCTION

A Wireless Sensor Network (WSN) consists of hundreds or thousands of tiny low-cost energy-limited nodes that have small capacities of sensing, processing and communicating via radio medium. Typically, these nodes report sensed data to a

base station for further processing. They are equipped with low-cost small-capacity batteries which are, in most cases, non-rechargeable and irreplaceable. Therefore, network lifetime is considered as an important issue for many applications such as: target tracking [1].

In contrast to sophisticated surveillance technologies such as RADARS, which are in fact reliable, robust and accurate but expensive, WSNs enable cheap technology that do not rely on any centralized infrastructure.

This technology which aims at performing and providing accurate data in a timely manner like traditional systems, brings-up new challenges related to data processing algorithms, communication systems and network organization. In many cases, cooperation between nodes helps solving these challenging open-issues.

On the contrary of single-node tracking systems, collaborative target tracking fuses data transmitted by many nodes and produces state-estimation of the target(s). However, these measurements are noisy, redundant and non-synchronized, and the inter-node communication is an energy-consuming task. Furthermore, neither reliable communication protocols nor complex data processing algorithms can be implemented on a sensor node because of its limited processing and communication capacities.

From this point of view, energy conservation in target tracking is a key issue and it can be achieved using different methods [2]–[4]. Prediction-based scheme coupled with selective activation of nodes is one of such methods. Figure 1 shows an example of a target tracking scenario in an event-driven WSN, where nodes are waked-up *on-demand* following the target path. Previous active nodes predict the activation zone in which not all nodes can effectively detect the target but only a subset of them. These active nodes collaborate between each other to generate an accurate estimation of the target state using a *in-network light-weight* data fusion algorithm. The gain of such algorithms is two-fold: (i) it generates state-estimates of the target, and (ii) it produces state-predictions

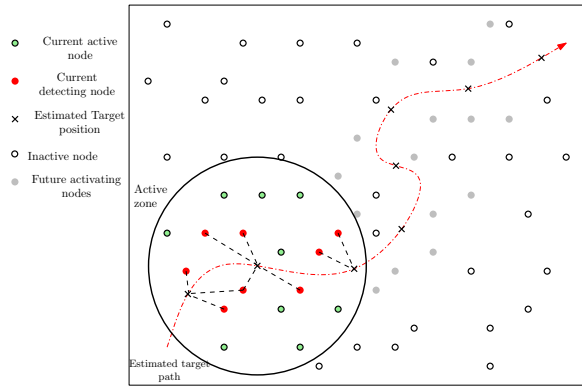


Fig. 1. Example of a Prediction-Based Scheme.

for the next sampling period.

Hence, these predictions serve as a basis for choosing nodes; this is called the *Sensor Selection Problem (SSP)* [5]. Roughly speaking, it consists at finding the subset of nodes with minimum cost that provide the maximum information utility among all the network's sensor nodes. In case of target tracking problem, the cost and the information utility can be defined respectively as: the energy consumption and the tracking data accuracy [6].

Another energy conservation technique consists of defining a schedule or a plane for nodes' activation and/or sleeping. Following the target trajectory predicted by the prediction algorithm, nodes are *waked-up* to execute some sensing and communication tasks for a determined period of time, and then they put themselves in a *deep-sleep* state. The other nodes that are not involved in the tracking task are put in the sleep state to preserve their energy resources. However, the schedule should not miss the target while it passes through the sensing zones of sensors and it should concern the ones with maximum energy resources.

This procedure requires collaboration between nodes and coordination between communication-related and sensing-related operations because of several reasons: first, the sensing measurements are redundant and noisy, and the multi-node target detection in contrast to single-node detection generates correlated measurements that can be fused. Second, the communication links are lossy which can be overcome by using collaborative protocols to deal with messages' loss. And finally, the dense nature of a WSN requires self-organization to reduce energy consumption. This can be achieved by structuring the network into clusters and/or trees that follows the target trajectory. In the following, we will draw a taxonomy of all these approaches to extract general recommendations for designing an energy-efficient target tracking scheme.

Nonetheless, we do not consider MAC-layer communication mechanisms such as *duty-cycling* [7] because they out of scope of this paper that is organized as follows:

In section II, we present our proposed classification with a brief description of each class. In section III, we describe the first class of schemes that are sensing-related. In section

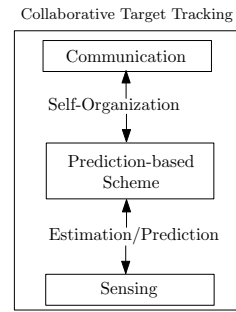


Fig. 2. Target Tracking Scheme Components.

IV, we study in more depth the other class of approaches that are related to the communication subsystem. In section V, we present another class of methods that do not fit into our proposed classification. We discuss and compare all these schemes in section VI and finally we conclude the paper in section VII.

## II. SCHEMES CLASSIFICATION

Before presenting our proposed classification, we give in the following subsection some definitions that help at characterizing a typical target tracking scheme.

### A. Target Tracking Schemes Characterization

Typically, a target tracking scheme consists of three subsystems, namely: the sensing subsystem, the estimation/prediction subsystem, and the communication subsystem. Figure 2 shows the relationship between the sensing and the communication subsystems. The estimation/prediction scheme extracts useful information from redundant and heterogeneous data transmitted by active nodes. It uses this information to extrapolate the target position in the future and then organizes the network to follow-up the target's path. The communication subsystem creates and maintains the cluster and/or the tree structure to minimize messages exchanges. In addition, the prediction algorithm predicts the target state for the next sampling period based on which, instants and durations of the sensing operations can be optimized.

In this paper, we consider that in an energy-efficient WSN-based target tracking scheme, all nodes should be initially in a sleep state, except nodes that are on the borders of the surveillance area. These nodes do the first target detection/identification operations, then they activate other nodes via *external activation* messages transmitted over a low-power channel called *paging channel*.

A tracking process is generally divided into successive tracking steps whose durations are constant or variable depending on the estimation/prediction algorithm. In each tracking step, the activation message is disseminated in a zone called *activation zone* whose range depends on the estimated target velocity and the measurements' error in the current tracking step.

After network initialization, the estimation/prediction algorithm computes a reliable estimation of the target state.

The filtering algorithm generates an estimation for the current tracking step and one or more predictions for the next tracking steps. If the target has a dynamic behavior during the current tracking step, then a cluster and/or a tree reorganization is triggered to follow-up the target trajectory. It is the current leader or the current root that generates target estimation and reports data to the sink.

A typical target tracking scheme should consider the following elements to be energy efficient:

- 1) **Quality of detection:** depending on the percentage of network coverage (which is related to the network initial deployment, nodes' sensing ranges, network density, etc.), target can be *watched* by one or more nodes that generate correlated readings about its state. A target tracking scheme should be able to measure the information utility of these data to decide about which nodes to select for the next tracking step? How long should be the activation range? How many nodes should be selected? etc. This measure helps also computing the current estimation error.
- 2) **Estimation/prediction algorithm:** the prediction algorithm should be *distributed* and *light-weight* depending on the equation state model of the target (linear or non-linear), the noise model of the sensors readings (Gaussian, non-Gaussian), the target sensing modalities (single modality, multiple modalities), and given the limited resources of sensor nodes, The Kalman Filter algorithm (KF) [8] is an accepted solution for estimation/prediction since it is easy to implement its distributed version such as in Distributed Kalman Filter (DKF) [9] and Kalman Consensus Filter (KCF) [10]. The algorithm is based on a two-step recursive procedure, namely: *update* step and *prediction* step and it converges for linear systems. However, for more complex equation state models, there exist other data filtering algorithm such as: Particle Filter (PF) [11], Variational Filter (VF) [12], Extended Kalman Filter (EKF) [13], Unscented Kalman Filter (UKF) [14], etc.
- 3) **Data reporting mechanism:** after target state estimation and prediction, choosing the data reporter node is another issue. Typically, when connectivity is provided, nodes that are close to the target with maximum energy resources should be selected. However, network reconfiguration may lead to a situation where the reporter node is far away from the sink and/or the target. Solutions such as the selection of backup reporter nodes or the establishment of a hybrid (static/dynamic) network structure can be applied.
- 4) **Activation mechanism:** the activation range depends on the target velocity. To avoid target loss, a multi-step activation mechanism with dynamic activation range can be applied. The activation plan can be static (pre-established at the beginning) or dynamic depending on the current estimated measurements' error.
- 5) **Logical network structure:** to optimize communica-

tions, a flat network structure is not the better solution. Clusters or trees can be temporarily constructed to localize the data fusion process. However, the target tracking scheme should tackle some problems related to the dynamic nature of WSN such as: leader election, cluster/tree reconfiguration, clusters boundary determination, etc.

## B. General Classification

Figure 3 depicts our proposed classification that is based on two complementary aspects regarding target tracking applications, namely: *sensing-related* and *communication-related* aspects.

In this general classification, we divide the sensing-related approaches into two subclasses: *Single-node Signal Processing* approaches (SP) and *Collaborative Signal Processing* approaches (CSP). The first class is out of scope of this paper. We are interested in the CSP class that is split into two other subclasses, namely: *information-driven* subclass and *data filtering* subclass. Information-driven techniques exploit the data content to optimize future readings. Whereas data filtering techniques generate precise information from noisy readings.

Communication-related approaches are also decomposed into two subclasses: *Routing/Aggregation* subclass and *Network Self-organization* subclass. Since routing and aggregation techniques are common to all WSN-based applications and they are well surveyed in the literature [15], we omit this subclass and we look to the other one which is directly related to target tracking applications. We divide it into three subclasses: *node selection*, *sleep scheduling*, and *dynamic clustering*.

Node selection techniques estimate nodes' resources and target detection probability of each node for the next sampling period. It estimates also nodes' contributions in the target *belief state* [16] and decide which node to activate and which one to put in the sleep state.

Since the sleep scheduling subclass is more important, it is also divided into three subclasses which are: *geometric-based*, *biological-based* and *coverage-based* techniques. These subclasses differ between each other according to the strategy adopted by each method to activate and deactivate nodes.

Note here that coverage-based approaches concentrate on preserving converge requirements while keeping nodes in the sleeping state. An in-depth description of each approach is given in sections III and IV.

An approach that merges Distributed Predictive Tracking (DPT) with regional Collaborative Signal Processing (CSP) and uses them alternately to track group of objects is proposed in [17]. It is based on the same concept that rules our classification.

## III. COLLABORATIVE SIGNAL PROCESSING METHODS

In collaborative signal processing techniques (called also distributed in-network data processing), instead of sending

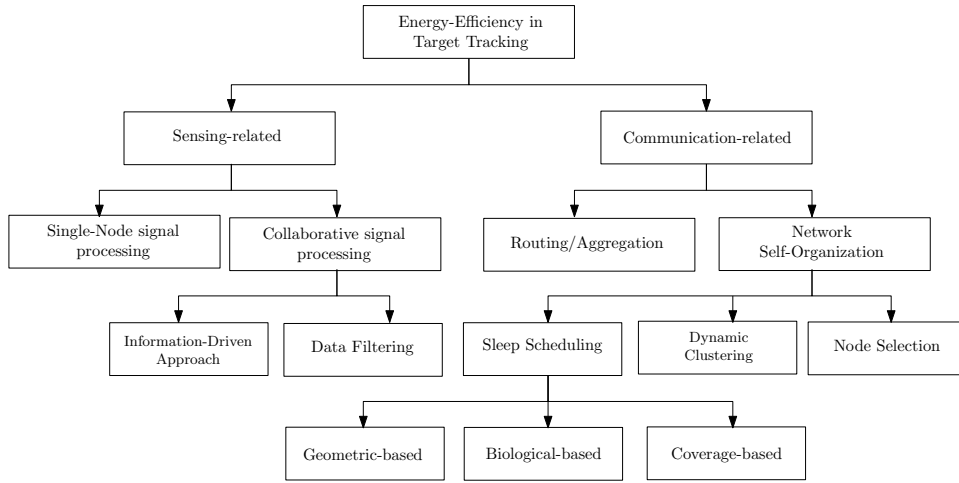


Fig. 3. General Classification.

data to the sink node to be processed by the end-user application, nodes collaborate between them to retrieve required information. They decide about which data to deliver and which one to aggregate or to compress. The goal here is to optimize network communications and reduce the number of nodes involved in the tracking process and the volume of messages exchanged between them. In the following subsections, two different techniques are shown: information-driven based techniques and data filtering based techniques.

#### A. Information-Driven Techniques

To our best knowledge, Information-Driven Sensor Querying (IDSQ) had been first proposed in [16]. The basic idea behind this approach is to explore the content of the data captured by sensors to optimize the future readings. Specifically, it aims at determining which sensor should take the measurements, and to whom it should send them. IDSQ requires collaboration among sensor nodes because the targets may have sparse spatiotemporal distribution. Given this hypothesis, target tracking in IDSQ is seen as a *sequential Bayesian estimation problem* where different measures of the information-utility of the sensor readings are proposed such as: Mahalanobis distance and entropy-based utility measure.

The activation mechanism defined in [18] selects an informative sensor such that the fusion of the selected sensor observation with the prior target location distribution would yield, on average, the greatest or nearly the greatest reduction in the *entropy* of the target location distribution. The problem addressed by the authors in [18] is how to efficiently evaluate the expected information gain attributable to each candidate node to selection without actually retrieving sensor data. They define the entropy-based heuristic to measure the quality of detection using the sensor's view about the target location (which is the geometric projection of the target location onto that of the sensor's observation perspective). This metric is a function of both target and sensor locations, and it is simpler than mutual-information method [19]. The main difference between the method proposed in [18] and the one in [20] is

that, the former involves only one sensor, whereas the later selects many sensors.

The Distributed Kalman Filter (DKF) algorithm with information-driven extension used as an estimation algorithm is proposed in [21]. R. Ofati-Saber showed in [21] that the common objective of improving individual information value of the sensors would force to perform an unplanned moving rendezvous near the mobile target. Collision avoidance between agents leads to a flocking behavior. He proposed a metric that measures the information value similar to the *Fisher Information* [22]. He showed also that adding the agent-target interaction to the flocking algorithm ([23] [24]) is a way of taking the information value of sensor measurements into account in motion planning of agents toward the target.

Another problem related to computational efficiency should be considered when we have to activate nodes with maximum information-utility. This problem can be non-trivial because the function of the next-step error covariance matrix can be non-convex. Authors in [25] propose to minimize the trace of the next step error covariance matrix to find the maximum of the function. They propose a relaxation approach that finds a computationally feasible sub-optimal solution. Thus, the trace of the next step error covariance matrix becomes a convex function and its minimum can be easily computed using convex optimization.

#### B. Data Filtering Techniques

With respect to the constraints of sensor nodes in terms of computation and communication, light-weight versions of classical filtering algorithms have been proposed. For example, in [10] the Kalman Consensus Filter (KCF) algorithm is proposed for nodes with limited sensing range. In this kind of network, not all the nodes of the network can observe the target but only a subset of them. The authors proposed a consensus-based filtering algorithm that is implemented over a logical P2P network of micro-filters. Each micro-filter is a *local estimator*. A high-level fusion center aggregates the state-estimates and the error covariance matrix of each micro-filter.

The KCF algorithm aims at reaching a consensus on estimates obtained by local KF rather than distributing the construction of the fused measurements and the covariance information of the central KF. The fusion center in KCF does not receive a large amount of data because of the hybrid architecture.

A cluster-based scheme that uses Extended Kalman Filter (EKF) as an estimation algorithm is also proposed in [26]. Tracking in [26] is done via successive selection of nodes. To select a node, the leader needs to know its *target detection probability* which can be deduced from the target state equations. Using this probability, one can compute the *Joint Detection Probability of all detecting sensors*. Unfortunately, this process is very complex and requires a Monte-Carlo simulation method. The authors propose a greedy approach in which, the sensors with high detection probability are selected.

Another aspect of EKF when used as an estimation/prediction algorithm is the problem of the Inter-Sensor Interferences problem (ISI). This problem appears with active sensors that track non-collaborative targets [27]. To resolve this problem, a time-division distributed technique is proposed in which each sensor senses the target alternatively within a predefined number of slots. The first variant of this technique is called the *periodic scheduling*: it is based on the division of time into periodic cycles each of which is assigned to a sensor (see figure 4).

If the scheduled sensor detects the target, it will calculate the difference between its measurement time and the previous time step and fuse its measurement with the existing target estimation using EKF. The problem of finding the minimum number of time-slots can be modeled as a *graph coloring problem* which is known as a NP-Complete problem. The second variants is called the *adaptive scheduling*. Its goal is to eliminate empty slots by scheduling the next tasking node for the next time step according to the predicted tracking accuracy which is derived from the trace of the covariance matrix of the state estimation using Unscented Kalman Filter (UKF). In figure 5, nodes  $s_2, s_3$  and  $s_4$  have empty detections because the target passes far from their sensing zones.

EKF can also be used to estimate the distance to the target by the current cluster-head. This has been proposed in [28] where detecting nodes send their measurements to a selected cluster-head which in turn sends a command message to the second nearest sensor to get initial coordinates of the target. After that, it executes a Least Square method to obtain a "good" estimation of the target position. The EKF algorithm is then triggered to estimate the current and the next target state. The target model in this scheme is Position-Velocity (PV) with 4 sensor distance measurements. The sampling period ( $\delta T$ ) is computed according to the target velocity.

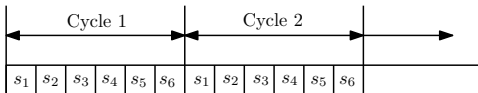


Fig. 4. Illustration of periodic scheduling.

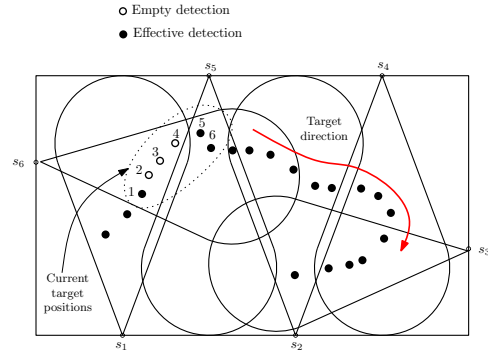


Fig. 5. Effect of empty slots according to [27].

When we use EKF as an estimation/prediction algorithm, it is important to find the most energy-efficient logical network topology (ex. a tree) that satisfies state estimation constraints. This problem is addressed in [29] where it is shown that choosing the tree with minimum energy consumption is very difficult. Thereby, the authors propose a tree reconfiguration algorithm composed of three procedures: (1) a recursive tree initialization procedure that uses the minimum power transmission to establish connections of each node with its immediate neighbors, (2) a switching tree topology procedure that is triggered when the desired quality is not achieved. It transforms some two-hop neighbors to one-hop neighbors, (3) a minimum energy subtree procedure that finds all the possible subtrees that satisfy the required quality and returns the one with minimum overall energy cost.

Moreover, it is also shown in [29] that minimizing the overall energy consumption may not lead to maximizing the network lifetime. Thus, a tree scheduling algorithm that choose  $M$  trees from  $N^{N-2}$  possible trees is necessary<sup>1</sup>. A linear programming solution has been proposed to solve this problem.

As stated above, the main objective of collaborative signal processing methods is to measure the quality of data that can be delivered by nodes to choose the most appropriate ones for activation. They focus on reducing readings' noise and predicting the target behavior in the future without paying attention to the network structure to reduce energy consumption in communications. According to our classification shown in figure 3, this the complementary aspect of sensing-related methods and we will deal it in the next section.

#### IV. NETWORK SELF-ORGANIZATION APPROACHES

The objective of network self-organization approaches is to extend the network lifetime by eliminating unnecessary wakening of nodes, planning rigorously their sleep state and adapting the network topology to the dynamic of the target. In the following subsections, we describe three subclasses that fit in this class of methods.

<sup>1</sup>Optimal scheduling of such trees is also known to be an NP-Complete problem.

### A. Sleep Scheduling

The presence of a target in the surveillance field is a *localized event* because of its limited observability and the nodes' limited sensing ranges. The energy emitted by the target is attenuated by spreading in space. Thus, only a subset of nodes that are located close to the target should be activated. The other nodes should stay inactive until they receive an explicit activation message from the previous tracking nodes indicating that the target is probably in their sensing ranges. The predictability of the target trajectory helps at determining, with a sufficient degree of confidence, which nodes to wake-up, at what time and for which duration. From this standpoint, sleep scheduling (or sleep planning) should consider several constraints such as: sensing and communication coverage, network lifetime, reliable detection, accurate tracking, etc.

In this subsection, we present the three subclasses of sleep scheduling methods, namely: biological-based, geometric-based and coverage-based.

1) *Biological-based Approaches*: they use bio-inspired concepts such as insect communities and gene programming notions.

One approach that treats the target as a virtual chemical emitter is proposed in [30]. It constructs influence contours whose strength decreases with distance from the target. Then, it selects the sampling period based on meta-data of the target using its *net traveled distance* (its past behavior). When the target enters the surveillance region, it is detected by the border nodes. Then, a group of nodes is pro-actively selected and assigned as a Main Node (MN) or a Helper Node (HN). The MN selects the next tracking group based on the predicted target state and the node's *centrality* which should be the largest.

The concept of Parallel Gene Expression Programming (P-GEP) is used in [31] to schedule nodes' sleeping. The target trajectory is modeled as a piecewise function which is divided into different shorter portions. This function is unknown before the target appears but it can be mined, and the future positions can be predicted from past location information. P-GEP includes a light-weight localization algorithm that uses the distances to the target to estimate the target location. During the trajectory mining process, some past location information are unnecessary, so they can be discarded using a *sliding window* mechanism whose size determines how many previous information is needed according to the prediction accuracy. This later is measured using the distance between the prediction location and the actual location. Given the prediction accuracy, an upper-bound and a lower-bound trajectories are computed. A fitness function is proposed to evaluate individuals in P-GEP: the higher the function value is, the better the individual is, i.e: the the prediction error is low. The node scheduling algorithm is based on a single-step or multi-step prediction model that uses the trajectory found to determine which nodes to wakeup at time  $t_{i+j}$  from the historical information up to time  $t_i$ .

2) *Coverage-based Approaches*: their goal is to preserve the event coverage requirements in the nodes' sleeping plane.

Different problems rise.

First, the problem of determining the optimal period length of activation in a sleep schedule of the *Controlled Greedy Sleep* algorithm (CGS) [32] which is addressed in [33]. This problem is modeled as a bi-partite graph whose properties determines the static and the dynamic *k-coverage* requirements. The basic idea behind this technique is that each node can estimate the number of neighbors that will benefit from its duty period, and based on this estimation, it decides to become active or not. The authors recommend to consider a short period of activation for dynamic networks.

Second, the off-duty eligibility rules problem which is addressed in [34]. The objective here is to identify redundant nodes to put them in off-duty mode without using any location information. The basic idea is that before scheduling nodes' sleeping, user application can specify the desired coverage percentage loss, then the corresponding threshold is determined using given expression or prior collected data pairs which case-depends: in the case of *the number of neighbors*, the threshold is the minimal neighbor's number, and in the case of *the nearest neighbor*, the threshold is the nearest neighbor distance, etc.

Last, the multi-node event watching problem i.e. TEK-WEM<sup>2</sup> which is formulated in [35]. The authors propose an algorithm that find the detection sets that satisfy the warning delivery delay and the network lifetime constraints. The algorithm uses a color-based method to construct all the *Breath-First-Search Trees* each of which is rooted at a gateway. Each tree corresponds to a detection set. The gateway adds greedily into the detection set sensors whose sensing components can help to k-monitor the atomic events. It can also add useless nodes for connectivity reasons. After that, the gateway builds the working schedule which is broadcast to all sensor nodes. A selection heuristic that takes into account the estimated lifetime, the number of helpful sensing components and the number of sensing components that equips the sensor is also discussed.

3) *Geometric-based Approaches*: these approaches use computational geometry to construct application-oriented network topology. They take profit from the nodes' location information to optimize target localization and tracking. In the following, we discuss three different schemes, each of them uses a different geometric concept.

Face-based Object Tracking (FOTP) scheme [36] uses a face-based architecture with a *hexagon* algorithm for prediction. It achieves energy efficiency by reducing the number of active faces as well as the number of waking nodes. A face is defined as "the subdivision of the maximal connected subset of the plane that does not contain a point on an edge or a node" [36]. The main idea behind FOTP is as follows:

First, some active nodes called "soldier nodes" detect the target then wake-up all the nodes in the face by which the object enters. In figure 6, the target enters by face  $F_1$  and nodes  $A$ ,  $B$  and  $N$  are waked-up to detect it. The nodes in the current

<sup>2</sup>Timely Energy-Efficient *k*-Watching Event Monitoring.

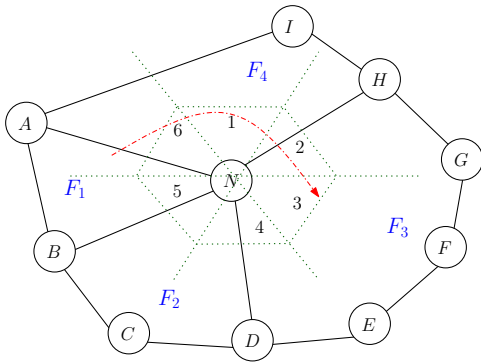


Fig. 6. Example of FOTP operations according to [36].

face can estimate the distance to the target. After that, they select the nearest node (NN) to the target as a leader. In the example (figure 6), node  $N$  is the leader. The NN determines by which edge the target enters and obtains its last position. Then, it computes its speed and direction. The current NN determines the next edge to which the target is going through (in figure 6 it is the edge  $N - H$ ) and selects the next NN from the set of nodes in the adjacent faces that intersects with the predicted edge. Candidate nodes select the next NN which is node  $F$  in this example. If the next NN cannot detect the target in *its time slot*, it sends a lost message to the last NN which will activate all its adjacent faces. If the target is not detected again, this last NN sends a message back to the base station.

Another concept used in geometric-based schemes is planarized graphs. A typical scheme that demonstrates this concept is Polygon-based Target Tracking scheme (PTT) [37], where a measure of the information-utility based on the Cramer-Rao Lower Bound (CRLB)<sup>3</sup> of the variance is used. The objective here is to minimize the number of nodes that participate in the tracking process. The algorithm is based on the *brink construction* procedure which consists at determining the critical region by connecting an edge called the *brink* to the active polygon. This critical region helps at confirming if the target is leaving the current polygon and entering another one, or not. The PTT scheme uses also a node selection procedure based on both the information-usefulness and the energy cost: before the target crosses the brink, a control message which contains the target state estimation is sent to the nodes in the forwarding polygon. The receiving nodes combine their measures with the received estimation to compute their weights. Each node can locally decide whether it should join the tracking operation or not, and thereby it determines its couple nodes.

Finally, *Kinematics* are used to reduce the active tracking area in multiple-target tracking [38]. Sensor nodes are classed into three categories: Boundary Nodes (BN), Worker Nodes (WN) and Computational Nodes (CN). The tracking area is mapped to a *Voronoi Diagram* and three different cases are

identified depending on the overlapped area of interest. Larger overlapped area of interest between two polygons results in a smaller number of sensors in that polygons, and vice versa.

### B. Node Selection

The second subclass of network self-organization approaches is nodes' selection which is directly related to network lifetime maximization since this later is often formalized as an optimization problem with constraints and its resolution leads to node selection program.

This problem has been modeled as a *knapsack problem* in [39]. The goal of the authors was to maximize the residual energy of the network while meeting the application QoS requirements. In their model, the authors subdivide the duration of the submitted task  $T$  into rounds of size  $t$ . The execution of the algorithm is alternated between different subsets of active nodes whose role will not change during the round. The algorithm seeks to select the best subset of nodes according to different objective function such as: minimum energy consumption, maximum residual energy, etc. According to this model, the nodes are the objects of potential relevance  $R_i$  and residual energy  $U_i$ . They have to be placed in of knapsack of capacity  $M$  which represents the energy budget. The energy cost of each node is its energy spent in sensing and communication activities. The other constraints of coverage, connectivity and QoS are included in the dynamic program that resolves the problem.

Network Lifetime Maximization Problem (NLMP) and Routing Path Length Minimization Problem (RPLMP) are jointly addressed in [40]. NLMP is expressed as the maximization of the number of sensor nodes which are kept in sleep state, and RPLMP is formalized as the minimization of the routing path length. The nodes that are turned on remain in this state until the end. These two problems are *proved to be NP-Complete* and three heuristics are proposed to solve it, namely: Naive Shortest Path Selection (NSPS), Dual Shortest Path Selection (DSPS) and Weighted Shortest Path Selection (WSPS). NSPS is better in minimizing the delay time while WSPS and DSPS are better in maximizing the network lifetime. The choice between these three heuristics depends on the application requirements.

Finding the upper bound of the network lifetime for any collaborative protocol is an interesting question that is addressed in [41]. The authors propose an approach based on *Role Assignment* in which three basic roles are defined, namely: sensing, gateway and aggregation. First, the problem is modeled as a linear program. Then, the constraints of the network topology and the sensing model are included. After that, it is transformed into a flow maximization problem in order to reduce computation complexity. To give a reliable estimation of the network lifetime upper-bound, energy consumption at the MAC layer should be incorporated into the proposed model.

Sensor management with respect to energy-efficiency is another alternative to select nodes. It uses many concepts such as: state-centric [42], rechargeable sensors [43] and target's

<sup>3</sup>CRLB is the inverse of the Fisher Information Matrix.



profile information [44], [45] to manage the sensing activity of the nodes.

For example, in [42] active nodes selection is based on a state-centric strategy. The prediction of the target state is computed whatever is the number of hops between the current data fusion center and the next one. In this scheme, the selection of the fusion center is based on the energy cost and all data transmissions are routed via minimal cost energy paths.

The activation range of sensors is also considered when we come to track a mobile target with large acceleration [44]. A Proportional-Integral-Derivative (PID) control system [46] is used to update activation range in each sampling period. The proposed algorithm measures the tracking quality and compares it with the required tracking quality. The average error determines the number of nodes to be activated and the activation range as well. The recovery process is based on the number of consecutive misses, the distance between the predictive positions, etc.

In [45], the Greedy-Selection Sensor Management (GSSM) scheme is proposed to assign a proper subset of sensors to track multiple targets. GSSM uses an information filter [47] for multi-sensor data fusion. Sensor management in this scheme is based on the fact that *not all* measurements contribute in improving the tracking accuracy<sup>4</sup>. Furthermore, information propagation uses two mechanisms to decide which subset of nodes receives the target information: (1) the *predicted subset mechanism* which is more optimal, but less-effective, and (2) the *nearest subset mechanism* which is sub-optimal but communication and delay-effective. In GSSM, the sensor management problem with the maximization of the sensors' information contributions is formulated as a binary optimization problem. The GSSM finds a near-optimal solution to this problem compared to the branch-and-bound algorithm. However, this later is more difficult to implement in a localized manner.

In [43], sensor nodes are supposed to be rechargeable using energy harvesting techniques. A set of scheduling techniques that take into account uncertainties of the energy incomes, and which are based on the ANSWER architecture [48] are proposed. The energy consumption depends on the current node states, namely: active state, inactive state, wakeup state, energy harvesting state and observation state. A *static active time* approach is proposed in which the complete duration of the day contains a regular alteration between active and inactive time. This approach increases the observation quality but suffers from the lack of adaptation to unexpected events. A *multi-parametric heuristic based* approach which depends on the currently stored energy, the probability of encountering an event, etc. is used to predict the length of the next active time. A third *utility-based active time* approach is proposed to overcome the problems of the heuristic approach.

Distributed Spanning Tree Algorithm (DSTA) is designed to track fast targets [49]. It generates successive predictions to wake-up nodes in  $t + \delta t$ ,  $t + 2\delta t$ ,  $t + 3\delta t$ , etc. where:

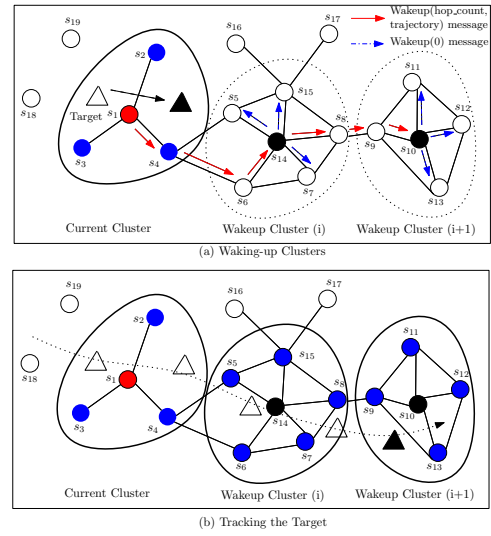


Fig. 7. Example of DSTA protocol operations according to [49].

$\delta t$  is the tracking step duration. The protocol is designed in two layers: the lower-layer where a spanning tree-based clustering protocol builds a cluster-based network structure (in figure 7 (a), three clusters are formed: the current cluster  $\{s_1, s_2, s_3, s_4\}$  with cluster-head:  $s_1$ , the wakeup cluster ( $i$ )  $\{s_5, s_6, s_7, s_8, s_{14}, s_{15}\}$  with cluster-head:  $s_{14}$  and the wakeup cluster ( $i + 1$ )  $\{s_9, s_{10}, s_{11}, s_{12}, s_{13}\}$  with cluster-head:  $s_{10}$ ), and an upper-layer where the wakening algorithm is executed. The clusters form a spanning tree rooted at the sink node. Due to the average speed of the target and its direction, clusters in the same direction are woken up. The number of these clusters depends on the target speed. The goal of this scheme is to decrease the probability of target loss.

The Global Prediction-based Algorithm (GPBA) [50] considers two main parameters: (1) the sampling duration and (2) the reporting frequency (to the sink node). It uses global profile rather than local profile because the global profile is accessible by all the nodes of the network. The mobile object (target) in GPBA can collect the information about its behavior from the network. Thus, nodes can use this information to activate a specific set of nodes. This approach is specific to objects tagged with a unique global ID. A learning phase is necessary to record the object movement frequency between nodes. Each node saves the frequency of each node by which the object is traversing. After that, based on the object profile, a *Tracking Leader* is elected as the current cluster-head that detects the target. The object profile is updated each time the object is lost.

### C. Dynamic Clustering

A WSN for target tracking is generally built on a cluster structure because of its aggregation and data fusion characteristics. In the literature, there exist pure dynamic clustering schemes as well as hybrid (static/dynamic) schemes. Cluster formation, maintaining, reconfiguration and cluster-head election are the main problems related to these schemes. In the

<sup>4</sup>This is called the Sensor Management Problem.

this section, we describe and discuss some protocols such as: ADCT, HTTP and HCTT, which have different methods to tackle the above-mentioned problems.

Starting with the Adaptive Dynamic Cluster-based Tracking (ADCT) protocol that is proposed in [51]. The cluster formation procedure is based on a two-phase mechanism. A broadcast phase and a notification phase. The node with the smallest distance/ID is chosen as a cluster-head. The sensor selection procedure is based on an optimal selection function which is a mixture of both data usefulness and energy cost. The nodes' usefulness can be deduced from the *bid messages* sent by members to the cluster-head. The cluster reconfiguration procedure is triggered when the predicted position of the target is on the boundary of the current cluster, in which case, the cluster-head sends a command message to the neighboring node nearest to the predicted position. Receiving nodes send an election message to their neighboring nodes and select the first replying one as a new cluster-head. The recovery mechanism is based on acknowledgment messages and waiting timers.

Clustering coupled with the Particle Filter (PF) is another scheme proposed in [52]. It is based on re-sampling method (SIR) to reduce the computation complexity of the PF by eliminating samples with small weights and preserving samples with big weights: that is called *bootstrap filter*. This approach suffers from degeneracy problem: the system may collapse to a single point. The solution proposed in [52] is a *local linearization* using EKF or UKF.

The utility function used in node selection is defined as the uncertainty of the target reduced by the additional measurements. It can be represented by the entropy of the belief state which can be used to select the best node among sensor candidates to maximize information gain. The cost includes: the bit rate between the cluster-head and the neighbor, the distance from the sensor node to the cluster-head and to the target, and the energy needed to receive one bit from neighbors. An optimization problem is formalized for which two scenario-dependent solutions were proposed: a *meta heuristic called GRASP* for static scenario and a *branch-and-bound* method for dynamic scenario.

Another protocol called Herd-Based Target Tracking Protocol (HTTP) is proposed in [53]. It is based on a three state-transition model, namely: sensing, sleeping and tracking. Each node computes its weight and decides to participate in the tracking process or not. Nodes that are in the tracking state form a cluster surrounding the target. The *backup herd node* is a node that has the same role as the *herd node* but it does not send data to the base station. The geographic region of the network is divided into virtual grids, each of which is monitored by a cluster-head. A node in the sensing state computes its weight periodically then it checks if it exceeds a specified threshold. Then, it goes to the tracking state. Note here that the node's weight depends on its distance to the target and the number of clusters that can participate in the tracking task is determined by the grid size. When the target moves out from the current grid to a new one, nodes within it can go either to the tracking state or to the sensing state depending

on the measurements data. Meanwhile, nodes of the old grid return back to the sleeping state because they cannot sense the target anymore. The herd reconfiguration is triggered based on the distance between the current herd head and the target.

Hybrid Cluster-based Target Tracking (HCTT) scheme [54] addresses the problem of cluster's boundary by integrating a dynamic on-demand clustering protocol and a static cluster-based target tracking scheme. According to [54], the boundary problem increases the tracking uncertainties and to solve it, HCTT checks whether there exist neighboring nodes that belong to another cluster or not. If yes, then these are boundary nodes and the cluster region is divided into three types: safety region, boundary region and alert region. Consequently, the dynamic cluster includes active boundary nodes that detect the target. The hand-off between static and dynamic clusters is based on the sensing data received from the nodes within these different regions.

The scheme proposed in [55] uses a backoff procedure to defer messages' broadcast, reduce the energy consumption and varying the transmission range to achieve data accuracy. The scheme consists of two processes: selection and release. In the selection process, nodes that detect the target simultaneously trigger the backoff procedure. The node with the lowest backoff period, broadcasts a *DETECT message* to its neighbors (nodes that are in its transmission range). Upon receiving this message, nodes stop their backoff procedures. The other nodes that are out of the transmission range and which has the lowest backoff time, do not receive the DETECT message. Thus, the node with the lowest backoff time will track the target. Collisions occur when two or more nodes transmit their DETECT message simultaneously because of their identical backoff time. In the release process, when the target moves out of the sensing range of the detecting nodes, a *RELEASE message* is transmitted to the neighbors. Upon receiving this message, nodes trigger a backoff procedure. Similarly, nodes that have the lowest backoff time will be selected to track the target.

## V. CLASSIFICATION OF SPECIFIC APPROACHES

In this section, we describe schemes that put special assumptions on the sensor capacities or handle special functionalities in the target tracking process or track special targets. These schemes do not fit into our classification proposed in section II because they use different methods to achieve energy-efficiency. The common aspect that relates this special classification and the above-described one is the ability of using prediction to optimize sensing and communication. In the following subsections we present the most important schemes that fit in this category.

### A. Continuous Object Tracking

Continuous objects contrary to single or individual objects, have a geometric shape and may expand in a large area such as: gas leaks, animal troupes, etc. Generally, this kind of objects cannot be represented by a single point or an atomic event. They often need multiple attributes to be described with. Thus,

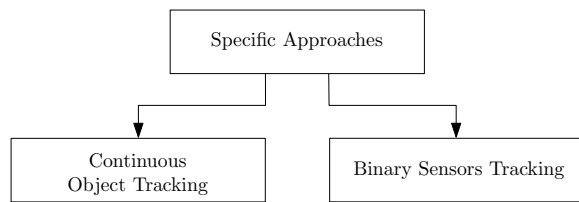


Fig. 8. Specific Approaches.

sensor nodes should have multiple sensing modalities to track such objects, i.e: they form a *Wireless Heterogeneous Sensor Network (WHSN)*. In WHSN, one sensor can detect multiple target's attributes. It is suitable for detecting and tracking composite events (ex. fire or pollution). A challenging task in tracking continuous objects is *event boundary determination*. Contrary to individual object tracking, in which the main problem is how to predict the next location of the target, and how to inform the next tasking nodes to take charge of the tracking task, composite events occupy a large area. The goal here is not to construct and maintain a network topology but to estimate attribute regions and to determine the event region.

According to [56], there exist four methods to track continuous objects:

- 1) All sensors in the phenomenon area report data to the sink.
- 2) Only nodes nearby the boundary area of the phenomenon are selected.
- 3) Nodes outside and inside the boundary area are selected (more nodes than in the previous method).
- 4) Few representative nodes that report the tracking data.

For example, in ECOT scheme [56], an adjustable sensing range technique is used where nodes may have five types: (1) undetect-to-detect, (2), detect-to-undetected, (3) border node, (4) representative node and (5) border point. Boundary detection is achieved via node sensing range adjustment: nodes of type 1 diminish their sensing range and nodes of type 2 increase their sensing range.

TOCOB is another scheme [57] in which each sensor wakes-up periodically and makes a local observation. A node becomes a CVN (*Changed Value Node*) when it observes a value in the current sampling period different from its last recorded value. This node broadcasts a COZ (*CompareOneZero*) message containing its ID and its status. A node becomes BN (*Boundary Node*) if it receives a COZ message with different value; It counts the COZ messages received during a period of time in order to decide to become an RN (*Representative Node*) or not. Contrary to the COBOM algorithm [58], not all nodes nearby the boundary will become BN but only those which receive different readings. RN selection is based on the number of COZ messages that are received by the BN nodes which determines the backoff time: the higher is the number of COZ messages, the shorter is the backoff time. Hence, nodes near the boundary will get high probability to become RN nodes.

CODA [59] uses a hybrid static/dynamic clustering ap-

proach by constructing a static-cluster backbone and determining boundary sensors which form after that a dynamic cluster to monitor the continuous object profile. CODA uses also the *Graham Scan algorithm* [60] to resolve the *Convex hull* problem i.e: determining the sensors located at the boundary of the cluster by the cluster-head. Boundary detection is based on the number of static clusters that detect the object. The cluster-head is notified via *sense* messages. Then, it executes the Graham Scan algorithm and organizes its boundary sensors within a dynamic cluster. When the object boundary moves out of the sensing range of the current boundary sensors, new clusters are formed.

In COLLECT [61], the accuracy of *event determination* is achieved by subdividing the estimated attribute regions into multiple non-overlapping faces. In each face, nodes determine whether the event has occurred or not. Some test rules such as AIT (*Alert-In Triangulation*), active role transition and passive role transition are used to design these procedures of event determination and border nodes selection. A simplified strategy of logical neighbors selection is used which is called: *short diagonal wins*. A more general design strategies are needed to prove the effectiveness of such technique.

### B. Tracking with Binary Sensors

Binary sensors generate one bit information indicating that the target is approaching or moving away, or it is present in the sensing range of the sensor or absent. These primitive sensors minimize the volume of data transmitted from sensors to the sink. However, the quality of tracking may degrade when noisy measurements and/or lossy links are present.

In [62], the authors propose a minimalist scheme of binary sensors that broadcast only a one bit of information to the base station to indicate that the target is approaching or moving away. A derived tracking algorithm based on particle filter is proposed. The algorithm generates a set of particles whose weights are computed based on the probabilities of moving from one point to another and the distances between sensors. The acceptance criterion of a particle is its associated probability which should be above a certain threshold.

Although this model is energy-efficient since it uses a small amount of data, it has the limit of the indistinguishableness of two targets close to each other or that move parallel to each other. Hence, an enhancement with a proximity bit is proposed to overcome this problem.

In [63], the problem of tracking a group of targets as a continuous region using binary proximity sensors is addressed. Sensors compute their probability of detection based on the duration of the target presence in their sensing field. They use this probability to determine if the target is really detected or not. In order to localize the group of targets, authors propose two algorithms to determine the monitoring region: the first one which is accurate but complex, computes the convex hull using the graham scan algorithm. However, the second one which compute the circle that contains the convex hull, is less complex but less accurate. Reporter node selection algorithm chooses the node closest to the plus (+) sensors. Based on the

TABLE I  
COMPARISON BETWEEN SCHEMES OBJECTIVES

Objectives	Sensing-related	Communication-related
Energy Consumption	Information-Utility measures Measurements' Error Data fusion & Compression	Selection of data reporting node Role assignment Multi-prediction Application-oriented topologies
Data Accuracy	Target Observability Sampling period length Sensor management	Network coverage & connectivity Geographic position information Topology reconfiguration

TABLE II  
COMPARISON BETWEEN MECHANISMS OF SENSING-RELATED SCHEMES

Mechanisms	[16]	[18]	[21]	[25]	[10]	[26]	[27]	[28]	[29]
Quality of detection	✓	✓	✓	✓	-	✓	-	-	-
Estimation/Prediction Algorithm	✓	-	✓	-	✓	✓	✓	✓	✓
Data Reporting Mechanism	-	-	-	-	-	-	-	-	-
Activation Mechanism	✓	✓	-	✓	-	✓	✓	✓	✓
Logical Network Structure	-	-	✓	-	✓	✓	-	-	✓

TABLE III  
COMPARISON BETWEEN MECHANISMS OF SLEEP SCHEDULING BASED SCHEMES

Mechanisms	[30]	[31]	[33]	[34]	[35]	[36]	[37]	[38]
Quality of detection	-	✓	✓	✓	✓	-	✓	-
Estimation/Prediction Algorithm	-	✓	-	-	-	✓	-	-
Data Reporting Mechanism	-	-	-	-	-	-	-	-
Activation Mechanism	✓	✓	✓	✓	✓	✓	✓	✓
Logical Network Structure	✓	-	-	-	✓	✓	✓	✓

TABLE IV  
COMPARISON BETWEEN MECHANISMS OF NODE SELECTION AND DYNAMIC CLUSTERING BASED SCHEMES

Mechanisms	[39]	[40]	[41]	[42]	[44]	[45]	[43]	[49]	[50]	[51]	[52]	[53]	[54]	[55]
Quality of detection	-	-	-	-	✓	✓	✓	-	✓	-	-	✓	-	-
Estimation/Prediction Algorithm	-	-	-	-	✓	-	-	✓	-	-	✓	-	-	-
Data Reporting Mechanism	-	-	✓	✓	✓	✓	-	-	✓	-	-	-	-	-
Activation Mechanism	✓	✓	-	✓	✓	-	-	✓	-	✓	✓	✓	-	✓
Logical Network Structure	✓	-	-	✓	-	-	-	✓	✓	✓	✓	✓	✓	✓

distance over which data are transmitted, and after iteration, the proposed algorithm gives the near-optimal coordinates of the reporter node. Finally, authors suggest a redeployment control algorithm when the target group moves away from the current one. This algorithm aims at reducing energy consumption by reducing the distances the data aggregations are transmitted.

Another binary proximity sensor tracking scheme is proposed in [12]. For non-Gaussian target state distribution, tracking becomes intractable. Instead of using a Particle Filter (PF), the authors propose *Variational Filtering* approach to reduce inter-cluster communications. In addition to this, a Binary Proximity Observation Model (BPOM) with a predefined threshold is used. This gives a solution to the problem with minimalist model. The advantage of VF over PF is the compression of the statistics required to update filtering distribution between successive instants. In order to reduce energy consumption, the authors adopt a Non-myopic cluster activation based on the prediction generated by the VF.

## VI. SCHEMES COMPARISON AND DISCUSSION

All the above-described schemes share the dual-objective of minimizing the energy consumption and improving the data accuracy:

- Energy-efficiency is related to the sensing and the communication operations.
- Data-accuracy can be expressed as the precision of the estimations or the amount of data extracted from the WSN given a certain network energy budget.

As it is shown in table I, on one hand, sensing-related energy-efficiency methods try to select nodes using a prediction algorithm with respect to coverage, connectivity and network lifetime constraints. They use data-fusion and data compression algorithm to minimize the volume of data to be forwarded. On the other hand, communication-related energy efficiency methods optimize routing and data reporting. They select reporter nodes based on the distance estimated between current active nodes and the target. They

use role-assignment and multi-step prediction approaches to balance energy consumption between nodes. Another objective of the network self-organization techniques is to construct application-oriented topology that helps at optimizing data-fusion process and extend the network lifetime.

Data accuracy is the opposite objective to the energy-efficiency. Based on our study, data accuracy depends on the network coverage and the target observability. Constructing an *efficient* topology with respect to coverage constraints and message communication cost is a challenging problem. Furthermore, the use of a minimalist model of target observation such as binary sensors, adds additional constraints but helps at reducing energy consumption. The specification of the information-utility measurements and the target profile definition are also key problems to enhance data accuracy.

Tables II, III and IV show the comparisons that we conducted between sensing-related, sleep scheduling-based and sensor selection and dynamic clustering-based mechanisms, respectively. It is easy to observe that few schemes consider *all* the energy-efficient aspects of a typical target tracking scheme. All the schemes address only two or three aspects (except two schemes [26], [44] that consider four aspects). Furthermore, the activation mechanism is the best part addressed by the different schemes followed by network logical structure and the estimation/prediction algorithm. Hence, more attention should be put on the other aspects that are *less-handled* such as data reporting.

As we can see on the tables, sensing-related schemes focus on the estimation/prediction algorithm in contrast to sleep scheduling based schemes and sensor selection and dynamic clustering based schemes that concentrate on the quality of detection and the logical network structure, respectively.

## VII. CONCLUSION AND FUTURE WORK

The energy problem in WSN is still an active research area where a huge body of research is produced. In this paper we surveyed some of the most recent target tracking schemes whose goal is to preserve the network energy and maintain an acceptable level of data accuracy. All the schemes that we described and discussed try to resolve the energy problem by allowing the interaction between the sensing layer and the communication layer and letting each layer to take profit from the other. However, we believe that the most energy-consuming layer is the communication layer and the optimization effort should be put on it.

In addition, we find that an important research track in geographic and geometric-based schemes is handling distances' and nodes' positions uncertainties in tree and/or cluster construction. Any error in nodes' coordinates may propagate during the computation process which may lead to wrong results. We believe that taking into account these uncertainties may improve the decisions about the target position predictions and the data reporting especially in multimedia WSN.

## REFERENCES

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.
- [3] N. Pantazis and D. Vergados, "A survey on power control issues in wireless sensor networks," *Communications Surveys & Tutorials, IEEE*, vol. 9, no. 4, pp. 86–107, 2007.
- [4] A. Aziz, Y. Sekercioglu, P. Fitzpatrick, and M. Ivanovich, "A Survey on Distributed Topology Control Techniques for Extending the Lifetime of Battery Powered Wireless Sensor Networks," *Communications Surveys & Tutorials, IEEE*, vol. PP, no. 99, pp. 1–24, 2012.
- [5] V. Isler and R. Bajcsy, "The sensor selection problem for bounded uncertainty sensing models," *Automation Science and Engineering, IEEE Transactions on*, vol. 3, no. 4, pp. 372–381, 2006.
- [6] W. Xiao, C. K. Tham, and S. K. Das, "Collaborative Sensing to Improve Information Quality for Target Tracking in Wireless Sensor Networks," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*. IEEE, 2010, pp. 99–104.
- [7] A. Bachir, M. Dohler, T. Watteyne, and K. Leung, "MAC essentials for wireless sensor networks," *Communications Surveys & Tutorials, IEEE*, vol. 12, no. 2, pp. 222–248, 2010.
- [8] G. Welch and G. Bishop, "An introduction to the Kalman filter," *University of North Carolina at Chapel Hill, Chapel Hill, NC*, 1995.
- [9] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *The IEEE 2000 Symposium On Adaptive Systems for Signal Processing, Communications, and Control 2000. AS-SPCC*. IEEE, 2002, pp. 153–158.
- [10] R. Olfati-Saber and N. F. Sandell, "Distributed Tracking in Sensor Networks with Limited Sensing Range," in *American Control Conference, 2008*. IEEE, Jun. 2008, pp. 3157–3162.
- [11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4. IEEE, 1995, pp. 1942–1948.
- [12] J. Teng, H. Snoussi, and C. Richard, "Decentralized Variational Filtering for Target Tracking in Binary Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 10, pp. 1465–1477, Oct. 2010.
- [13] R. Meinhold and N. Singpurwalla, "Understanding the Kalman filter," *American Statistician*, vol. 37, no. 2, pp. 123–127, 1983.
- [14] J. Candy, "Model-based signal processing," *The Journal of the Acoustical Society of America*, vol. 119, p. 2553, 2006.
- [15] R. Rajagopalan and P. Varshney, "Data-aggregation techniques in sensor networks: a survey," *Communications Surveys & Tutorials, IEEE*, vol. 8, no. 4, pp. 48–63, 2006.
- [16] F. Zhao, J. Shin, and J. Reich, "Information-Driven Dynamic Sensor Collaboration," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, Mar. 2002.
- [17] D. Benferhat and J. F. Myoupo, "A Physical CPT and Regional CSP-Based Hybrid Algorithm for Energy Efficiency in Target Tracking in Wireless Sensor Networks," in *2010 Ninth International Conference on Networks*, 2010, pp. 127–132.
- [18] H. Wang, G. Pottie, K. Yao, and D. Estrin, "Entropy-based Sensor Selection Heuristic for Target Tracking," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, Apr. 2004, pp. 36–45.
- [19] E. Ertin, J. Fisher, and L. Potter, "Maximum mutual information principle for dynamic sensor query problems," in *Information Processing in Sensor Networks*. Springer, 2003, pp. 558–558.
- [20] W. Zhao, Y. Han, H. Wu, and L. Zhang, "Weighted Distance Based Sensor Selection for Target Tracking in Wireless Sensor Networks," *Signal Processing Letters, IEEE*, vol. 16, no. 08, pp. 647–650, Aug. 2009.
- [21] R. Olfati-Saber, "Distributed Tracking for Mobile Sensor Networks with Information-Driven Mobility," in *Proceedings of the 2007 American Control Conference*, Jul. 2007, pp. 4606–4612.
- [22] J. Passerieux and D. Van Cappel, "Optimal observer maneuver for bearings-only tracking," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 34, no. 3, pp. 777–788, 1998.

- [23] C. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. ACM, 1987, pp. 25–34.
- [24] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 401–420, 2006.
- [25] J. E. Weimer, B. Sinopoli, and B. H. Krogh, "Relaxation Approach to Dynamic Sensor Selection in Large-Scale Wireless Networks," *28th International Conference on Distributed Computing Systems Workshops*, pp. 501–506, 2008.
- [26] J. Lin, W. Xiao, F. L. Lewis, and L. Xie, "Energy-Efficient Distributed Adaptive Multisensor Scheduling for Target Tracking in Wireless Sensor Networks," *Instrumentation and Measurement, IEEE Transactions on*, vol. 58, no. 06, pp. 1886–1896, Jun. 2009.
- [27] X. Wen-Dong, W. Jian-Kang, X. Li-Hua, and D. Liang, "Sensor Scheduling for Target Tracking in Networks of Active Sensors," *ACTA Automatica Sinica*, vol. 32, no. 6, pp. 922–928, Nov. 2006.
- [28] B. Liu, F. Ren, C. Lin, and X. Jiang, "Performance Analysis of Sleep Scheduling Schemes in Sensor Networks Using Stochastic Petri Net," pp. 4278–4283, 2008.
- [29] L. Shi, A. Capponi, K. Johansson, and R. Murray, "Resource optimisation in a wireless sensor network with guaranteed estimator performance," *IET Control Theory and Applications*, vol. 04, no. 05, pp. 710–723, 2010.
- [30] Y. E. M. Hamouda and C. Phillips, "Metadata-Based Adaptive Sampling for Energy-Efficient Collaborative Target Tracking in Wireless Sensor Networks," in *10th IEEE International Conference on Computer and Information Technology*, 2010, pp. 313–320.
- [31] S. Dai, C. Tang, S. Qiao, Y. Wang, H. Li, and C. Li, "An Energy-Efficient Tracking Algorithm based on Gene Expression Programming in Wireless Sensor Networks," in *The 1st International Conference on Information Science and Engineering*, 2009, pp. 774–777.
- [32] G. Simon, L. Gonczy, and B. Cousin, "Dependable k-coverage algorithms for sensor networks," in *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*. IEEE, 2007, pp. 1–6.
- [33] G. Bergmann, M. Molnár, L. Gönczy, and B. Cousin, "Optimal Period Length for the CGS Sensor Network Scheduling Algorithm," in *Sixth International Conference on Networking and Services*. IEEE, 2010, pp. 192–199.
- [34] D. Tian and N. D. Georganas, "Location and calculation-free node-scheduling schemes in large wireless sensor networks," *Ad Hoc Networks*, vol. 2, no. 1, pp. 65–85, 2004.
- [35] X. R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part III: Motion Models of Ballistic and Space Targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 01, pp. 96–119, Jan. 2010.
- [36] X. Ji, Y.-Y. Zhang, S. Hussain, D.-X. Jin, E.-M. Lee, and M.-S. Park, "FOTP: Face-based Object Tracking Protocol in Wireless Sensor Network," in *Fourth International Conference on Computer Sciences and Convergence Information Technology*, 2009.
- [37] M. Z. A. Bhuiyan, G. Wang, and J. Wu, "Polygon-Based Tracking Framework in Surveillance Wireless Sensor Networks," in *15th International Conference on Parallel and Distributed Systems*, 2009, pp. 174–181.
- [38] A. A. U. Rahman, M. Naznin, and M. A. I. Mollah, "Energy-efficient Multiple Targets Tracking Using Target Kinematics in Wireless Sensor Networks," in *Fourth International Conference on Sensor Technologies and Applications*, 2010, pp. 275–280.
- [39] F. Delicato, F. Protti, L. Pirmez, and J. F. de Rezenda, "An Efficient Heuristic for Selecting Active Nodes in Wireless Sensor Networks," *Computer Networks*, vol. 50, no. 18, pp. 3701–3720, 2006.
- [40] L. Liu, H. Li, J. Wang, L. Li, and C. Li, "Heuristics for Mobile Object Tracking Problem in Wireless Sensor Networks," *Frontiers in Algorithms*, pp. 251–260, 2009.
- [41] M. Bhardwaj and A. P. Chandrakasan, "Bounding the Lifetime of Sensor Networks via optimal Role Assignments," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2002, pp. 1587–1596.
- [42] J. Lin, L. Xie, and W. Xiao, "State-Centric Multi-Sensor Scheduling for Target Tracking in Wireless Sensor Networks," in *Information, Communications and Signal Processing, 2007 6th International Conference on*. IEEE, 2008, pp. 1–4.
- [43] V. Pryma, D. Turgut, and L. Bölöni, "Active time scheduling for rechargeable sensor networks," *Computer Networks*, vol. 54, no. 4, pp. 631–640, 2010.
- [44] C.-C. Chen, J.-M. Hsu, and C.-H. Liao, "HAMA: A Three-Layered Architecture for Integrating Object Tracking and Location Management in Wireless Sensor Networks," in *Third International Conference on Multimedia and Ubiquitous Engineering*, 2009, pp. 268–275.
- [45] Q. Ling, Y. Fu, and Z. Tian, "Localized sensor management for multi-target tracking in wireless sensor networks," *Information Fusion*, pp. 194–201, 2011.
- [46] K. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *Control Systems Technology, IEEE Transactions on*, vol. 13, no. 4, pp. 559–576, 2005.
- [47] T. Vercauteren and X. Wang, "Decentralized sigma-point information filters for target tracking in collaborative sensor networks," *IEEE Transactions on Signal processing*, vol. 53, pp. 2997–3009, 2005.
- [48] S. Olariu, M. Eltoweissy, and M. Younis, "ANSWER: AutoNomous netWorked sEnsoR system," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 111–124, 2007.
- [49] A. Alaybeyoglu, O. Dagdeviren, A. Kantarci, and K. Erciyes, "A Distributed Wakening Based Target Tracking Protocol for Wireless Sensor Networks," in *Ninth International Symposium on Parallel and Distributed Computing*, 2010, pp. 165–172.
- [50] O. Garcia, A. Quintero, and S. Pierre, "A global profile-based algorithm for energy minimization in object tracking sensor networks," *Computer Communications*, vol. 33, pp. 736–744, 2010.
- [51] W. Yang, Z. Fu, J. Kim, and M. Park, "An adaptive dynamic cluster-based protocol for target tracking in wireless sensor networks," in *Proceedings of the joint 9th Asia-Pacific web and 8th international conference on web-age information management conference on Advances in data and web management*. Springer-Verlag, 2007, pp. 157–167.
- [52] L. Arienzo and M. Longo, "Energy-Efficient Target Tracking in Sensor Networks," *Ad Hoc Networks*, pp. 249–264, 2010.
- [53] X. Xing, G. Wang, and J. Wu, "Herd-Based Target Tracking Protocol in Wireless Sensor Networks," *Wireless Algorithms, Systems, and Applications*, pp. 135–148, 2009.
- [54] Z. Wang, W. Lou, Z. Wang, J. Ma, and H. Chen, "A novel mobility management scheme for target tracking in cluster-based sensor networks," *Distributed Computing in Sensor Systems*, pp. 172–186, 2010.
- [55] K. Jang, "Location Tracking for Wireless Sensor Networks," *Next Generation Teletraffic and Wired/Wireless Advanced Networking*, pp. 306–315, 2007.
- [56] D.-X. Jin, S. H. Chauhdary, X. Ji, Y.-Y. Zhang, D.-H. Lee, and M.-S. Park, "Energy-Efficiency Continuous Object Tracking Via Automatically Adjusting Sensing Range in Wireless Sensor Network," in *Fourth International Conference on Computer Sciences and Convergence Information Technology*, 2009.
- [57] J. Kim, K. Kim, C. Hussain, M. Cui, and M. Park, "Energy-Efficient Tracking of Continuous Objects in Wireless Sensor Networks," *Ubiquitous Intelligence and Computing*, pp. 323–337, 2010.
- [58] C. Zhong and M. Worboys, "Energy-Efficient Continuous Boundary Monitoring in Sensor Networks," Technical Report, 2007. Available: <http://ilab1.korea.ac.kr/papers/ref2.pdf>, Tech. Rep., 2007.
- [59] W.-R. Chang, H.-T. Lin, and Z.-Z. Cheng, "CODA: A Continuous Object Detection and Tracking Algorithm for Wireless Ad Hoc Sensor Networks," in *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*. IEEE, 2008, pp. 168–174.
- [60] B. Chazelle, "Approximation and decomposition of shapes," *Advances in Robotics*, vol. 1, 1985.
- [61] K.-P. Shih, S.-S. Wang, H.-C. Chen, and P.-H. Yang, "COLLECT: Collaborative event detection and tracking in wireless heterogeneous sensor networks," *Computer Communications*, vol. 31, pp. 3124–3136, 2008.
- [62] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a Moving Object with a Binary Sensor Network," in *Proceeding of SenSys'03*. ACM, Nov. 2003, pp. 150–161.
- [63] D. Cao, B. Jin, S. K. Das, and J. Cao, "On collaborative tracking of a target group using binary proximity sensors," *Journal of Parallel and Distributed Computing*, vol. 70, pp. 825–838, 2010.