Selecting Tracking Principals with Epoch Awareness

Oliviu Ghica*
Dept. of Electrical Engineering and Computer Science
Northwestern University
Evanston, Il

Goce Trajcevski*
Dept. of Electrical Engineering and Computer Science
Northwestern University
Evanston, Il

Fan Zhou†
Dept. of Computer Science
U. Elec. Sci and Tech., China
Chengdu, P.R. China

oliver,g-trajcevski,fan-zhou@northwestern.edu

Roberto Tamassia‡
Dept. of Computer Science
Brown University
Providence, RI
rt@cs.brown.edu

Peter Scheuermann*
Dept. of Electrical Engineering and Computer Science
Northwestern University
Evanston, Il
peters@eecs.northwestern.edu

ABSTRACT

This work addresses the problem of principal node selection during the tracking process in Wireless Sensor Networks (WSNs). In a typical tracking scenario, the location of a mobile unit is determined via collaborative trilateration by the nodes that have the tracked object within their sensing range. One of the participants in the trilateraion—the tracking principal—is in charge of transmitting the location and time information to a designated sink. However, as the moving object changes its location, a new principal needs to be determined and handed off the task of the subsequent sensing, trilateration and transmission to the sink. We observe that in many WSN applications in which sensing/sampling needs to be combined with multihop transmission and, possibly, in-network aggregation, the typical processing is organized in synchronized intervals, called epochs. We postulate that taking the semantics of the epoch into consideration is important when selecting tracking principals and we present efficient algorithmic solutions towards this goal. Our experiments demonstrate that the proposed approach can yield significant reduction in the number of hand-offs between consecutive tracking principals, when compared to previous works.

*Research supported in part by NSF grant CNS-0910952

†Work performed while visiting Electrical Engineering and Computer Science Department at Northwestern University

‡Research supported in part by NSF grant CCF-0830149

ACM GIS '10, November 2-5, 2010, San Jose, CA, USA
(c) 2010 ACM ISBN 978-1-4503-0428-3/10/11 . . . $10.00

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication; Distributed Networks

General Terms

Algorithms, Experimentation, Performance

Keywords

Epoch Awareness, Tracking, Sensor Networks

1. INTRODUCTION AND MOTIVATION

A large number of GIS applications rely on some form of *location based services* (LBS) [21], for which the efficient management of the *location-in-time* information pertaining to mobile entities is of paramount importance. The management of such information, along with efficient query processing techniques, is studied in the field of Moving Objects Databases (MOD) [9], where one of the challenges is obtaining the data stream of the object's whereabouts [7, 18]. While many vehicles are equipped with on-board devices [20] to obtain GPS-based location information, in many settings (e.g., limited GPS coverage in downtown areas or rural areas), location information needs to be determined using other type of sensing equipment and/or inter-vehicle communication [15].

Due to their ability to self-organize in a network along with the sensing and computational capabilities of individual nodes, wireless sensor networks (WSNs) have been extensively studied and used in variety of GIS-related application domains, including traffic management, environmental monitoring, and emergency response [10, 13, 22]. One of the canonical problems in WSN settings is the *tracking* of mobile objects and its related issues, including improving the accuracy of the tracking process and trading off the quality of the tracking information for energy savings [1, 4, 19, 23, 24, 26, 32].

We consider scenarios in which tracking is performed via the

cluster-based tracking protocol, such as the one proposed in [5]. A designated node, the cluster head, assumes the role of *tracking principal* (or, simply, *principal*) and is in charge of coordinating the tracking process and managing tracking information, such as identity and location, of the target moving object at a given time. The tracking principal relies on range measurements from other cluster members to determine the object's location, via trilateration [27]. In addition to managing the collaborative trilateration computation and initiating the transmission towards the designated sink-node, the current tracking principal has another important task. Namely, it participates in identifying the *next* tracking principal, when necessary, and transfer to it any accrued tracking information in order to continue coordinating the tracking process in close proximity of the moving target. Such transfers should be kept to a minimum, however, due to the: (1) overhead associated to the accrued tracking information that needs to be transferred and (2) delays incurred in the process of forming a new cluster during which the principal may lose track of the moving object.

At the heart of the motivation for our work is the observation that many query-processing tasks in WSN settings are organized in discretely-synchronized time-intervals called *epochs*, during which the (multi-hop) routing is combined with some in-network aggregation [16]. Epochs typically consist of the following phases (cf. Figure 1):

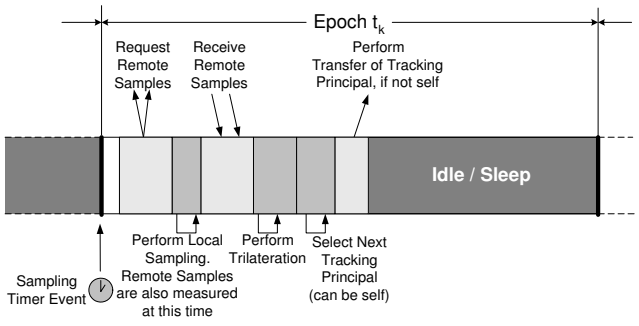
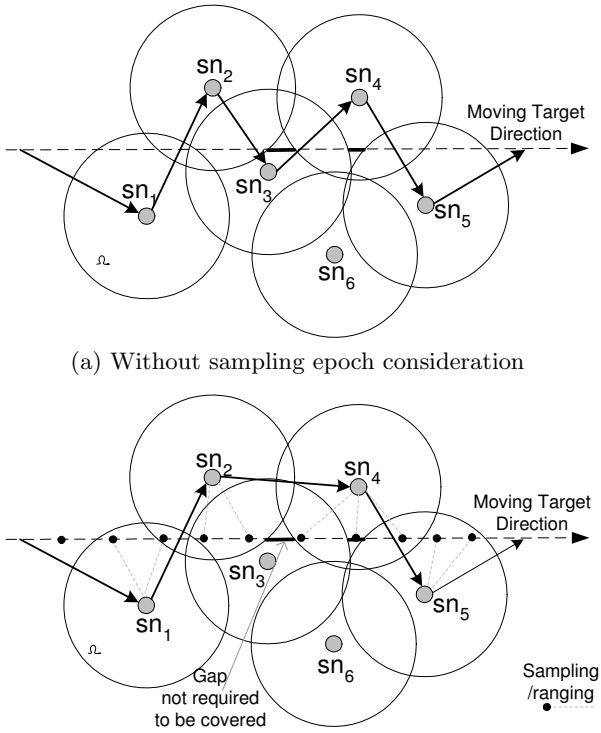


Figure 1: Lifecycle of a tracking epoch

1. Request range measurements from certain cluster members (e.g., via multicast)
2. Perform local range measurement
3. Collect range measurements from requested cluster members
4. Estimate the object's location via trilateration of the acquired range measurements
5. Select the next tracking principal, if necessary, and transfer tracking information to it
6. Idle/sleep for the remaining of the sampling epoch

The method of He and Hou [11] generates a sequence of principal nodes such that the union of their respective sensing-coverage disks completely covers the (expected) trajectory of a moving target while maintaining a minimal area of intersection. However, it does so by assuming *time-continuity* of the sampling process. For example, according to the methodology presented in [11], the selection of tracking principals will yield, as illustrated in Figure 2(a), the sequence of nodes $\{sn_1, sn_2, sn_3, sn_4, sn_5\}$.



(a) Without sampling epoch consideration

(b) With sampling epoch consideration

Figure 2: Motivation

However, given the discrete epoch-nature of the target sampling, the coverage is needed only at *discrete* time-instants. Figure 2(b) illustrates an alternative, shorter sequence of 4 principals $\{sn_1, sn_2, sn_4, sn_5\}$ that yields effectively the same *observed* coverage, leveraging upon the *coverage gap* (bold lines) between two consecutive samples (bold dots).

Adopting the notation from He and Hou [11], let L denote the length of the trajectory of a moving target, and let $\tilde{D}$ represent the effective range at which, with high probability and in a manner in which certain levels of tracking accuracy are satisfied, a candidate sensor node can be found with respect to the current principal node. He and Hou give the following lower bound on the average number of tracking principals required:

$$\left\lceil \frac{L}{1.1\tilde{D}} \right\rceil .$$

The goal of this work is to further improve upon this bound, by taking into consideration the semantics of the sampling epochs, relative to:

- $\tilde{v}$ – the expected object's motion (velocity),
- I – the duration of the sampling epoch,
- $NB(sn_i)$ – the set of available 1-hop neighbors of the current tracking principal sn_i.

Intuitively, we aim at employing a function:

$$\zeta(sn_j, NB(sn_i), \tilde{v}, I)$$

to base upon the selection of the next principal, where $sn_j \in NB(sn_i)$ is one of the 1-hop neighbors of sn_i.

This function will be used by the current tracking principal to select the next tracking principal from among its neighboring nodes. The function is actually a local heuristics which captures the discrete nature of the sampling process and, in addition, incorporates the fact that within a given epoch there may be other temporal aspects, besides the sampling itself, that need to be considered. Along these lines, the main contributions of this work are as follows:

- We propose a new methodology for selecting the tracking principals that incorporates the *discrete-timing* aspects of the sampling epochs into the tracking process.
- We develop an efficient algorithm for implementing the proposed methodology.
- We present extensive experimental evaluations of our approach, demonstrating that it can yield a significant reduction in the number of principal hand-offs during the tracking process in WSN.

In the rest of this paper, Section 2 gives preliminary background. Our principal selection mechanism is presented in Section 3, followed by the experimental observations in Section 4. In Section 5, we position our work in the context of related literature and in Section 6, we conclude and outline directions for future work.

2. PRELIMINARIES

In this Section we describe the basic notations used in this paper, and overview two previously proposed mechanisms for principal-selection to which we compare our method.

Let $SN = \{sn_1, sn_2, \ldots, sn_N\}$ denote the set of nodes in a given WSN. Each node sn_k knows its location (x_k, y_k) via GPS or other techniques, e.g., beacons [17]. Nodes are assumed to be static, know the locations of their one-hop neighbors and are capable of detecting an object within the range of sensing R_s, e.g., based on vibration, acoustics/echo, etc. [12]. The network is assumed to be dense enough to ensure communication coverage [28]. For a node $sn_i \in SN$, the set $NB(sn_i) = \{sn_j \in SN | \ \|sn_i, sn_j\| \leq R_c\}$, specifies the neighbors within its communication range R_c, where $\|sn_i, sn_j\|$ represents the Euclidean distance between nodes.

For a moving object o_i, $P_{k,o_i} \in SN$ denotes the tracking principal at time t_k. The *trilateration*-based (observed) location of the moving object o_i at time t_k is denoted by $\hat{L}_{k,o_i}$. In the sequel, for simplicity, we omit the o_i subscript when there is no ambiguity.

2.1 Closest Proximity Selection

The Closest Proximity based Selection (CPS) of a tracking principal is a naive approach, in which the subsequent tracking principal P_{k+1} at time t_{k+1} is selected based on its proximity to the target's *currently detected* location $\hat{L}_k$, i.e.:

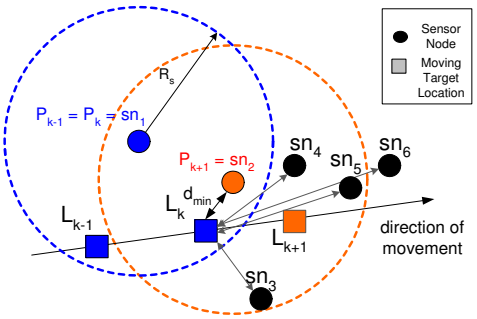


Figure 3: The Closest Proximity Leader Selection. Node sn_1 is the current tracking principal at time t_k and $\hat{L}_k$ is the trilaterated location of the moving target. Node sn_2 is selected as the tracking principal for the next sampling epoch t_{k+1} since it is physically closer to $\hat{L}_k$ than any other nodes in $NB(sn_1)$.

$P_{k+1} = \text{argmin}_j(\|sn_j, \hat{L}_k\|)$, where $sn_j \in NB(P_k)$.

Figure 3 illustrates the CPS methodology. While this selection mechanism performs poorly for reducing the number of required principals, it has the potential advantage of increased robustness (cf. Section 4).

2.2 Relay Area Based Selection

The *Relay Area Based* (RAB) principal selection methodology proposed by He and Hou [11] targets the lower bound of the required sequence of principal nodes for target tracking. Unlike CPS, RAB is a *predictive* approach, in the sense that the selection of the principal is based both on the current location $\hat{L}_k$ of the target, and its predicted *velocity*. The RAB method determines an annular sector, called *relay-area*, where the next principal node should ideally reside, as illustrated in Figure 4. The relay area is determined by three tunable parameters:

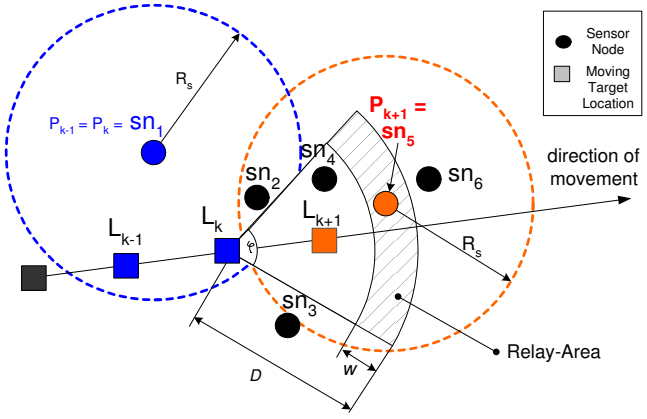


Figure 4: The Relay Area Based leader selection (RAB). When target o_i approaches the limits of the sensing area (L_k) of the current principal $P_k = sn_1$, based on velocity information, node sn_5 is selected as the next principal P_{k+1}, which is located within the relay area.

- w – width of the annular sector-area;
- φ – angle of the annular sector; and
- $\tilde{D}$ – the radius of the outer-circle of the annular sector.

The relay-area is iteratively expanded until it will comprise at least one sensor node, which will be subsequently selected as the next tracking principal.

3. TRACKING PRINCIPAL SELECTION WITH SAMPLING LOOK-AHEAD

We now present our method for the selection of tracking principals. We introduce the concept of *effective* coverage with epoch-awareness, and subsequently explaining our proposed technique, which leverages upon it.

3.1 Spatio-Temporal Coverage

As an object moves, its trajectory intersects the sensing/coverage area of a given node throughout a particular time-interval; we refer to such spatial intersection segments and the corresponding time-interval as the *spatio-temporal coverage* of a given node with respect to the mobile object.

Let $[t_k^-, t_k^+]$ denote the interval of *active* tracking within the k-th epoch, just *after* a particular node, say, sn_j, has been selected as a tracking principal. The time-instant t_k in the k-th epoch when the actual sampling is performed must satisfy $t_k^- \leq t_k \leq t_k^+$. In other words, t_k^- is the first time-instant, *after* the $(k-1)$-th epoch, in which sn_j samples under the new role of selected principal. We note that it is not necessarily the case that a tracking principal needs to be changed with every new epoch. In the example of Figure 5, node sn_j is a tracking principle in two consecutive epochs.

Segment $\overline{AD}$ in Figure 5, represents the "nominal" coverage of the node sn_j, however, its *actual* coverage consists only of the samples that are within segment $\overline{CD}$ (denoted as L_k and L_{k+1}). Hence, the time-portion of the coverage where object's trajectory corresponds to segment $\overline{AC}$ represents a *blind* coverage for all practical purposes. Understanding the role of the blind coverage is one of the contributions of this work. The time interval of the spatial-temporal coverage of a given node sn_j for a certain object's trajectory can be divided into the following components:

- *a priori blind* coverage, denoted as T_c^-, is the interval during which the mobile target is within the sensing range of sn_j, but sn_j has *not* yet been selected as tracking principal (e.g., segment $\overline{AB}$ in Figure 5).
- *a posteriori blind* coverage, denoted as T_c^+, consists of the interval between the time instant at which sn_j has been activated as a tracking principal, up to the first location-sampling (e.g., segment $\overline{BC}$ in Figure 5).
- *effective* coverage, denoted as T_c^e, is the remaining portion of the moving object's trajectory being tracked by sn_j (e.g., $\overline{CD}$ in Figure 5).

We note that the *blind* coverage components are of relevance during the "hand-off" process, where a new node is being selected as the next tracking principal. The complete time-interval of the spatio-temporal coverage of a given principal node can be specified as:

$$T_c = T_c^- + T_c^+ + T_c^e \leq 2R_s/\hat{v}_{max},$$

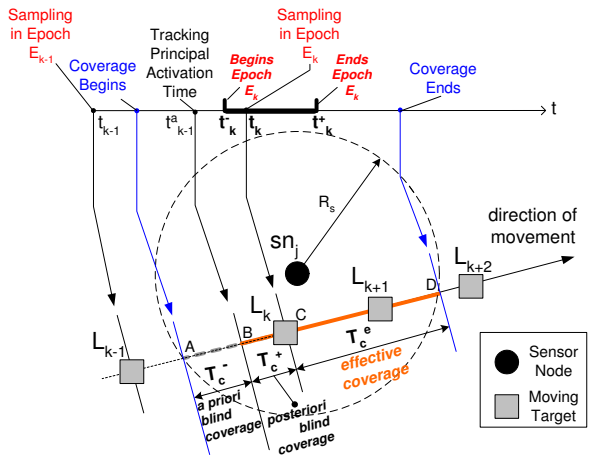


Figure 5: Blind vs. effective coverage

where $\hat{v}_{max}$ denotes the object's expected maximal speed.

While the RAB method aims at increasing T_c^e by maximizing the nominal coverage T_c (and implicitly reducing the a priori blind coverage T_c^-), the main difference with our work is that we also incorporate the role of T_c^+ into the principal selection process. As our experiments will demonstrate, this approach can yield savings in the number of hand-offs between successive principals.

3.2 Sampling Look-Ahead Selection

Similarly to the RAB approach, our *Sampling Look-ahead Selection* methodology (SLS) relies on location-prediction. However, in addition to considering the *expected trajectory*, we incorporate the *expected locations* and the duration of the sampling epochs, denoted as I. Using a buffer containing W of the most recent *(location,time)* samples, we use the following formula to determine the expected location $\hat{L}_k$ at the beginning of the k-th epoch:

$$\hat{L}_k(x_k, y_k) = \begin{cases} \hat{x}_k = \hat{x}_{k-1} + \sum_{i=k-W+1}^{i=k-1} w_i(\hat{x}_i - \hat{x}_{i-1}), \\ \hat{y}_k = \hat{y}_{k-1} + \sum_{i=k-W+1}^{i=k-1} w_i(\hat{y}_i - \hat{y}_{i-1}) \end{cases} \quad (1)$$

We assume an application-dependent "weight-factor" w_i assigned to each known past displacement of the moving target. Under the "continuous" spatio-temporal coverage, assuming $W = 2$ and $w_1 = w_2$, the optimal coordinates $\hat{P}_k$ of the next tracking principal that achieves minimal blind-coverage during the next transfer is given by the formula:

$$\hat{P}_k(x_k, y_k) = (x_k + R_s\cos(\varphi), y_k + R_s\sin(\varphi)) \quad (2)$$

where φ is the angle between the X-axis of the reference coordinate system and a vector parallel to the velocity vector $\vec{v}$ of the moving object (cf. Figure 6). Note that the $\hat{L}_k(x_k, y_k)$-coordinates correspond to the expected location of the moving target at the *next* sampling instant. Ideally, it should coincide with the sampling instant t_k of epoch $[t_k^-, t_k^+]$ under the new principal P_k. If this can be achieved, then we minimize the blind coverage and, at the same time, maximize the *coverage gap*, i.e., the spatial segment (and the corresponding temporal interval) during which no sampling should occur, as illustrated in Figure 6.

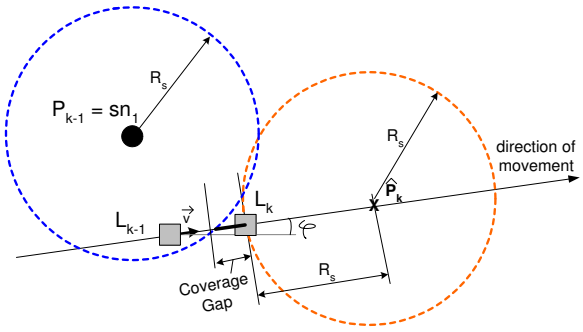


Figure 6: Coverage Gap

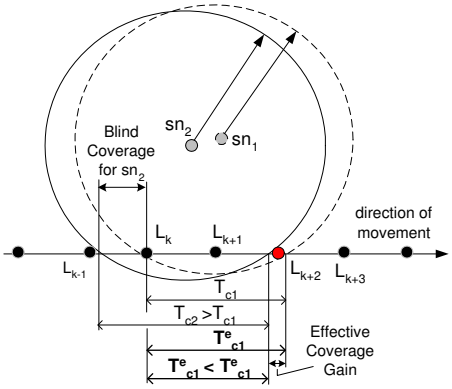


Figure 7: Discrete sampling vs. continuous coverage

However, in practice, the locations of the available sensor nodes in the neighborhood of the previous principal need not coincide with the ideal location $\hat{P}_k$. Consequently, the actual locations of the nodes need to be incorporated into the methodology for selecting the next tracking principal. A feature of our work is that, in addition to the actual physical locations of the sensor nodes, we also address the minimization of the $T_c^- + T_c^+$ portion – the blind coverage. We specifically consider the impact of the actual location-sampling time on the principal selection process.

With reference to the example of Figure 7 involving two candidate nodes, we make the following observations:

1. The sn_2 may appear to be a better candidate to assume the role of tracking principal under a *continuous* coverage process as in RAB, because it exhibits larger nominal coverage: $T_{c2} > T_{c1}$;
2. However, taking into consideration the *discrete* nature of the sampling process via epochs, which is the case with SLS, we observe that the object is already "too deep" into the coverage disk of sn_2 when the sampling of the k-th epoch occurs and the spatial coverage beforehand is not exploited (blind coverage).
3. SLS accounts for the blind coverage and selects sn_1 as principal node for which the object is "just enough" into its coverage disk at the *time-instant* when k-th epoch's sampling occurs. By reducing the coverage of the gap-area, sn_2 gains additional effective coverage for an additional sampling point. Thus, sn_1's larger effective coverage $T_{c1}^e > T_{c2}^e$ allows it to cover 3 consecutive epochs, as opposed to 2 epochs for sn_2. Hence sn_1 is, in fact, a better candidate.

Let sn_i denote the current tracking principal, which needs to perform a "hand-off" for the subsequent epoch, and $\mathcal{D}(sn_i, R_s)$ denote the disk centered at the location of the sensor node sn_i with radius R_s (the sensing radius). From the set of its neighbors $NB(sn_i)$, we need to select one, say, s_j, for which the segment given by the intersection of the (expected) trajectory of the tracked moving object with $\mathcal{D}(sn_j, R_s)$ yields the largest number of sampling-points along it. For this purpose, we define a function as follows:

$$\zeta(sn_j, NB(sn_j), \tilde{v}, I) = \\ |\ \{\hat{L} | \hat{L} \text{ is a sampling point} \in \mathcal{D}(sn_j, R_s)\}\ | \quad (3)$$

Function $\zeta(sn_j, NB(sn_j), \tilde{v}, I)$ calculates the expected number of samples covered by node $sn_j \in NB(sn_i)$ in the subsequent sampling epochs. The selection process determines the neighbor(s) of sn_i which covers the maximum number of samples. As for the right-hand side of the Equation 3, it is evaluated based on the following parameters:

- $\tilde{v}I$ – the expected displacement of the moving target during a sampling epoch;
- $T_{c_j} \cdot \tilde{v}$ – the nominal *spatial* coverage of node sn_j with respect to the moving target's trajectory;
- L_A – the location of the moving object where nominal coverage begins (cf. point A in Figure 5); L_A is determined as the first intersection point of the boundary of $\mathcal{D}(sn_j, R_s)$ with the moving object's trajectory;
- L_{k+1} – the first effective sample covered by next principal sn_j based on notations in Figure 5.

then the expected number of samples covered by node $sn_j \in NB(sn_i)$ can be expressed as :

$$\zeta(sn_j, NB(sn_i), \tilde{v}, I) = \left\lceil \frac{T_{c_j} \cdot \tilde{v} - \|L_A, L_{k+1}\|}{\tilde{v}I} \right\rceil \quad (4)$$

where the expression $\|L_A, L_{k+1}\| = (T_c^- + T_c^+) \cdot \tilde{v}$ represents the compensating factor of the blind coverage.

The implementation of the SLS selection, relying on the function $\zeta(sn_j, NB(sn_i), \tilde{v}, I)$, is formalized in Algorithm 1.

Algorithm 1 iterates over the set of neighboring nodes in $NB(sn_i)$ (cf. lines 3-5), which are potential *candidates* for resuming the role of tracking principal, and stores the best candidates (cf. lines 8-9) into set NB_{cand}. The NB_{cand} set consists of the nodes expected to cover the maximal number of location samples, based on the expected motion of the tracked object. We re-iterate that Algorithm 1 is a heuristic, which is reflected in the fact that line 8 of the algorithm relies on the function $\zeta(sn_j, NB(sn_i), \tilde{v}, I)$ when deciding which neighbor-node should be kept as a candidate, and the function itself relies on the *expected* sampling-locations.

Algorithm 1 SLS principal selection

Input:
sn_i: current tracking principal
Output:
The identity of the next tracking principal

1: $NB_{temp} = NB(sn_i)$;
2: $NB_{cand} = \emptyset$ // initialization
3: **while** $|NB_{temp}| \neq \emptyset$ **do**
4: Select $sn_j \in NB_{temp}$;
5: $NB_{temp} = NB_{temp} \setminus \{sn_j\}$
6: **if** $NB_{cand} = \emptyset$ **then**
7: $NB_{cand} \leftarrow NB_{cand} \cup \{sn_j\}$
8: **else if** $\exists\ sn_q \in NB_{cand}$ such that $\zeta(sn_j, NB(sn_i), \tilde{v}, I) > \zeta(sn_q, NB(sn_i), \tilde{v}, I)$ **then**
9: $NB_{cand} \leftarrow (NB_{cand} \setminus \{sn_q\}) \cup \{sn_j\}$
10: **end if**
11: **end while**
12: Pick a node from NB_{cand} as the next principal

We now analyze some important properties of Algorithm 1. First, we address the time complexity of our SLS approach. For each node $sn_j \in NB(sn_i)$, a constant number of operations are performed to detect the length of the intersection segment of the object's trajectory with $\mathcal{D}(sn_j, R_s)$, and to count the number of location samples along it. Subsequently, the current-principal (sn_i) uses Algorithm 1 to select the next principal. Hence, the time complexity is $O(NB(sn_i))$, linear in the number of the neighbors of sn_i.

When it comes to the message complexity, we note that the entire decision process is done locally by sn_i (the current principal) itself. The only SLS-specific message overhead occurs due to the actual hand-off of the tracking data from the current principal to the next one[1]. A hand-off message contains accrued application-specific tracking information, such as moving object's identity (o_i) and possibly a buffer of (location, time) trajectory data, such as in [24].

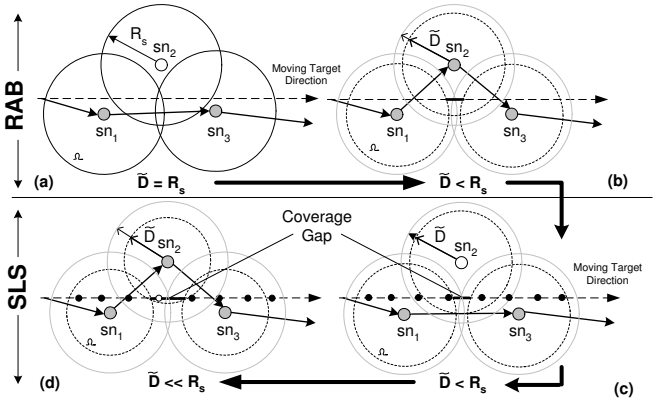


Figure 8: Impact of the effective sensing range

We conclude by discussing how to break ties when there are multiple candidate nodes for the subsequent role of tracking principal (line 12 of Algorithm 1). The first approach is to randomly select a node in NB_{cand}. Noting that Algorithm 1 relies on the *expected* location(s) of the tracked object, an alternative method is to have an extra "future-window" as the criterion for selecting the tracking principal. Specifically, we could recursively extend line 12, by applying the *while* loop (line 3-11) to each node in NB_{cand}, and select the one with the highest expected future pay-off. We defer a detailed theoretical evaluation of this heuristic for the future work.

3.3 On the Effective Sensing Range

In theory, the coverage gap is dictated by two factors: (1) the sensing range R_s and (2) the expected displacement of a moving target during one sampling epoch: $\delta = \tilde{v}I$.

In *practice*, however, the coverage gap may be much larger. Even if a target is within the sensing range of a given sensor, the *precision* of the localization may not be accurate due to signal attenuation and noise. While a full analysis based on various noise models is beyond the scope of this work, it is generally recognized (cf. [11]) that the levels of measurement imprecision increase proportionally with distance, i.e., ranging of closer targets yields better localization accuracy. Therefore, when certain Quality of Monitoring (QoM) thresholds are user-specified, the sensing range[2] is effectively reduced to $\tilde{D} \leq R_s$. It is not always the case that QoM are known a priori, and may be user-specified at run-time via a tracking query, whereupon the nodes may consequently adjust their effective sensing range dynamically.

However, even with this consideration, the main advantage of SLS, when compared to the existing approaches, is preserved. Figure 8 illustrates the differences between RAB and SLS as the effective sensing range is lowered. In the depicted scenario, under ideal conditions in which $\tilde{D} = R_s$ (cf. Figure 8 (a)), both RAB and SLS yield the same selection of a sequence of two tracking principals. If $\tilde{D}$ is slightly lower than R_s, RAB defaults to a sequence of three tracking principals (c.f. Figure 8 (b)) whereas SLS maintains a sequence of two principals in its selections (c.f. Figure 8 (c)). As it can be seen, it takes a much higher decrease in the effective sensing range ($\tilde{D} << R_s$), as in Figure 8 (d), for SLS to end up with a sequence of three principals.

4. EXPERIMENTS

Our experiments were performed using the open-source, SIDnet-SWANS simulator for WSN [8]. The network consists of 750 homogeneous nodes with simulated ranging capabilities that implement the equivalent of an active ultrasonic echo ranging system, running on a standard MAC802.15.4 link layer protocol.

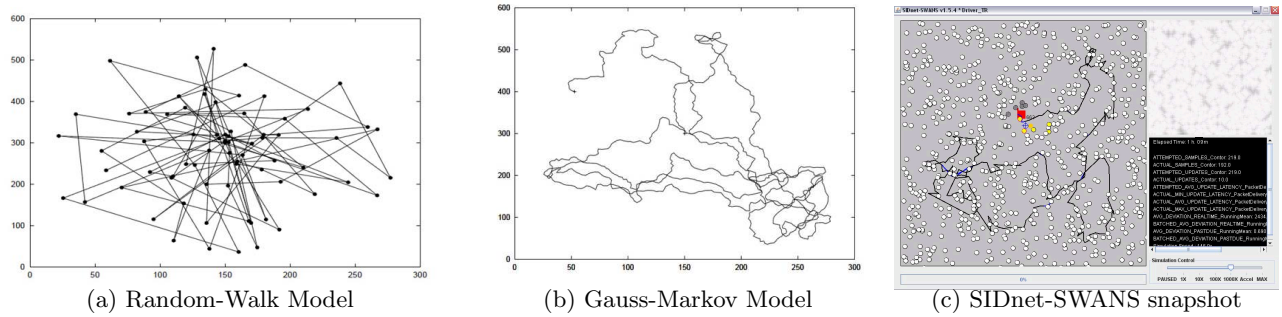
While mobility models such as *random walk* or *random waypoint* are often used, one of their drawbacks is the lack of *spatio-temporal dependency* [2]. To overcome this, we used the *Gauss-Markov Mobility Model* (GMMM) [3, 14] which

[1]We do not include the overheads due to neighborhood discovery during the initial network configuration, as this is a typical cost related to topology establishment. We also note that the trilateration cost is independent of the SLS methodology and is a standard part of any tracking protocol.

[2]We note that, in practice, the effective sensing range $\tilde{D}$ will be actually used in expression (4) when determining the location L_A, instead of R_s

Table 1: Leader Selection Experiments Table

Communication/ Sensing Range [m]	*Effective* Sensing Range $\tilde{D}$[m]	Principal Selection Methodology	Node Density [neighbors/node] / Field Area [m^2]	Object Type / Average Speed [mph] / Expected Epoch Displacement [m] / Corresponding Sampling Interval [s]	Trajectory Window [# vertices]
R_c=90 m /R_s = 45 m	45 m	CPS	$8 / 1,300^2$	"Walk" / 4mph / 9-18m / SI= 4-8s	W = 2
	40 m	RAB	$12 / 1,150^2$	"Bike" / 10mph / 9-18m / SI= 1.5-3s	
	35 m	SLS	$16 / 1,000^2$	"Car" / 25mph / 9-18m / SI= 0.8-1.5s	



(a) Random-Walk Model (b) Gauss-Markov Model (c) SIDnet-SWANS snapshot

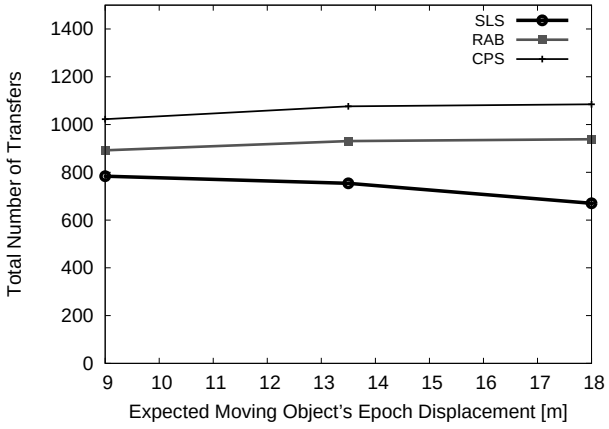
Figure 9: Comparison of GMMM model vs. Random-Walk model

does exhibit spatial and temporal dependency. As illustrated in Figure 9 (based on [3]), GMMM models yield traces that are more similar, especially in terms of sinuosity [6] to real traces. Essentially, GMMM works on a time-slot basis where at each slot the speed and direction are computed based on the ones from the previous time-slot. Figure 9(c) illustrates a snapshot of SIDnet-SWANS Simulator performing tracking on a GMMM-based mobile object.

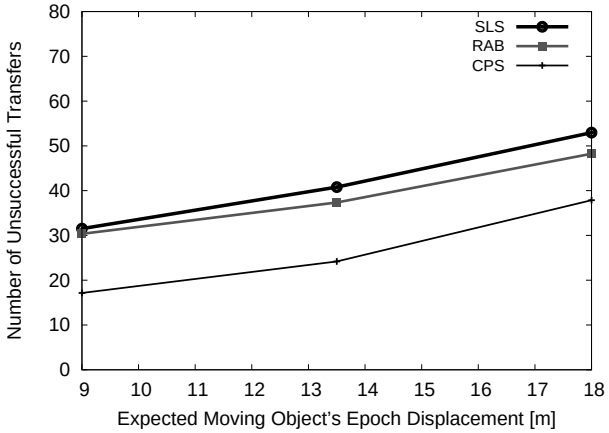
The experimental configuration space is summarized in Table 1. Accordingly, the main parameters of interest are: (1) the *moving target's displacement* $\delta = vI$ during a sampling epoch, which is dictated by the average speed of the specific type of mobile object as well as the value of the sampling interval; (2) the effective sensing range to account for possible measurement noise; and (3) the average number of 1-hop neighbors. The sliding trajectory buffer that we use for location prediction has a size of two vertices (W = 2). Traces were generated to be representative for three common types of moving objects: walks (people), bikes and cars, calibrated according to the average speed of the real-traces that we had available: 4, 10 and 25 *mph* respectively. Each experiment spans 3 hours of simulation time and consists of two parts: (1) bootstrapping and neighbor discovery protocols in SIDnet-SWANS; and (2) the actual tracking, in the remaining hours.

4.1 Impact of Moving Object Displacement

The first results that we report concern the influence of the spatial displacement that is exhibited by the moving object during an epoch. The spatial displacement, from the SLS perspective, dictates the maximum size of the coverage gap $T_g = T_c^- + T_c^+ \leq \delta$ that can be relied upon as a potential source for improvement. To this end, w.r.t. the effective sensing range of 35-45 meters, we have selected sampling intervals of 4-8 seconds for slow moving objects, such as human-walks, and .8-1.5 seconds for fast moving objects, such as motorized vehicles. This selection is based on a two-fold desideratum: (1) from an energy-saving perspective,

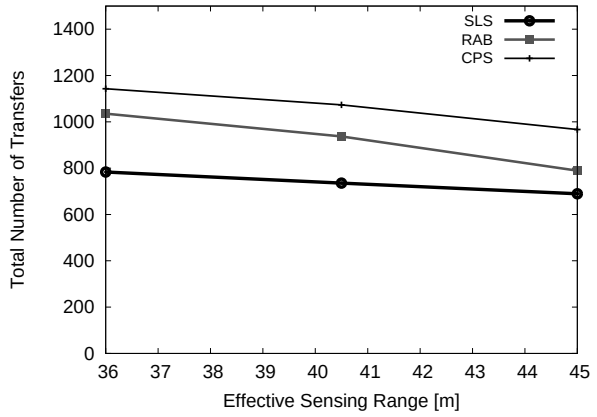


(a) Total Number of Leader Transfers

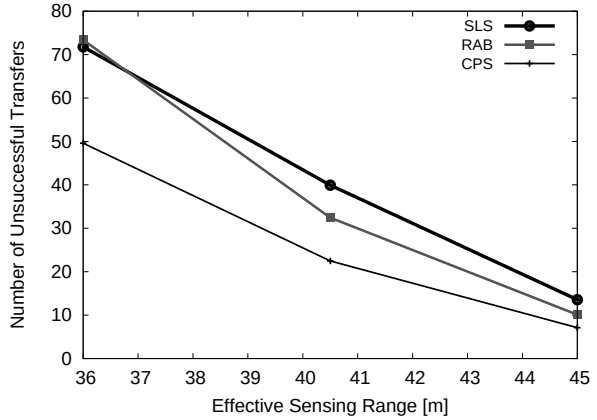


(b) Total Number of Unsuccessful Leader Transfers

Figure 10: Impact of Moving Object Epoch's Displacement

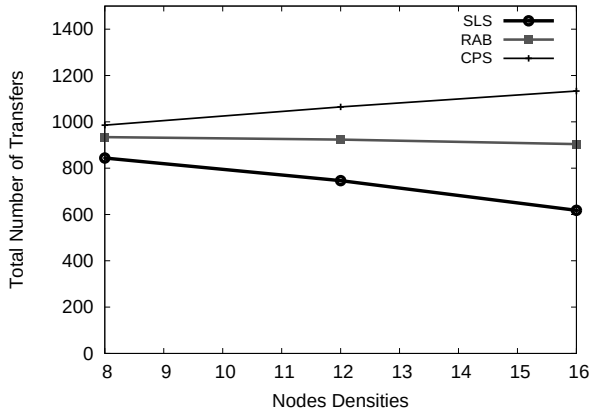


(a) Total Number of Leader Transfers

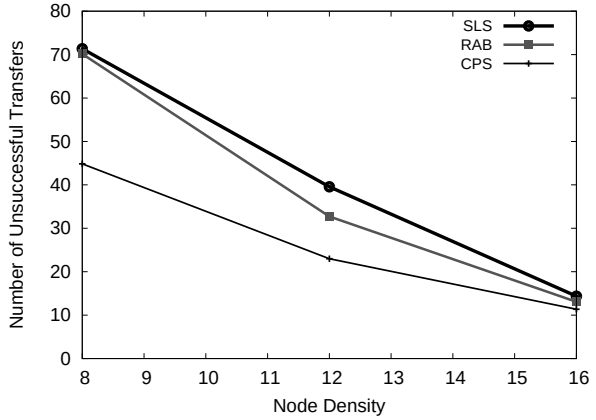


(b) Total Number of Unsuccessful Leader Transfers

Figure 11: Impact of the Effective Sensing Range



(a) Total Number of Leader Transfers



(b) Total Number of Unsuccessful Leader Transfers

Figure 12: Impact of Nodes' Density

large-sampling intervals are desired; (2) from a reliability of the tracking process standpoint, frequent sampling (small sampling intervals) are ideal.

Figure 10(a) demonstrates that a 12%-28% improvement can be achieved by SLS when compared to RAB and 28%-38% when compared to CPS, as the potential coverage gap increases from 9m to 18m. Larger coverage gaps, corresponding to larger sampling intervals, negatively affect the number of transfers in both CPS and RAB – this is because, with larger coverage gap, the effective coverage $T_c^e = T_c - (T_c^- + T_c^+)$ is lowered, hence the transfers will take place more often. SLS compensates for this to some extent through the coverage gap, attempting to pick nodes that are further away along the trajectory of the mobile target. However, according to Figure 10(b), the risk of an unsuccessful transfer $P_k \rightarrow P_{k+1}$ also increases. This is because larger coverage gaps increase the risk of miss-predicting the future locations of the moving object during a principal transfer and the object may not end up being covered by the predicted transferee. For example, the percentage of unsuccessful transfers (relative to the total number of attempted transfers) increases from by 3% to 8% for larger sampling intervals.

4.2 Impact of Effective Sensing Range

We tested three different effective sensing ranges in order to check whether certain trend can be established. Smaller ranges translate into more frequent principal hand-offs. As illustrated in Figure 11(a), RAB is quite more sensitive to the changes in the sensing range than SLS. The performance gain of SLS is maintained to 12%-24% when compared to RAB, and 28%-31% when compared to CPS.

From a reliability standpoint, a smaller coverage area increases the risk of target deviation from its predicted trajectory and bypass the coverage area of the intended principal, resulting in a wasted hand-off due to the inability of the principal to continue tracking. The fail-safe mechanism for unsuccessful transfers consists in principal election among nodes that detect the object in their vicinity.Figure 11(b) illustrates the relative ranking of the three approaches that we considered, and an interesting observation is that the sensitivity of RAB to the effective range makes it perform marginally worse than SLS when it comes to the absolute number of failed transfers under the lowest sensing range. This is because RAB starts with a more limited set of candidate nodes as the relay area is bounded by $\tilde{D}$, resulting in poor selections overall when the effective range is lower.

4.3 Impact of Nodes Density

Figure 12(a) illustrates the strong impact of the nodes density, expressed as the average number of 1-hop neighbors in the network, on the total number of tracking principals. As shown, in the settings of low nodes densities, the SLS based selection can only provide 7% improvement over RAB. However, as the density increases, the result shows around 19% improvement over RAB for moderately dense networks of 12 neighbors per node, and up to 30% on topologies with 16 neighbors per node on average. When it comes to CPS performance, it is interesting to observe a degradation which is due to the fact that, in high-density networks, CPS can find nodes much closer to the current location of the moving target, which would qualify as the next-principal. This, in turn, causes a lot more hand-offs to be performed.

As for the reliability, the number of failed transfers reduces inversely with the average number of neighbors, RAB exhibiting between 5% and 10% fewer such transfers than SLS, as it can be observed in Figure 12(b).

In conclusion, we have experimentally demonstrated that taking into consideration the specifics of the tracking epoch can provide significant additional savings in terms of the number of tracking principals at an average cost of 5%-10% increase in unsuccessful transfers. We note, however, that the ratio of unsuccessful transfers to the total attempted transfers is very small, below 5% overall. CPS exhibits the lowest-performance due to its overly conservative selection, however, it is the most robust in terms of ensuring successful transfers.

5. RELATED WORK

The problem of tracking in WSN settings is considered to be a canonical one, and many works have addressed various aspects of it, including the issue of leader election during the tracking process.

We reiterate that the closest approach to ours addressing the problem of tracking principal hand-off in a leader-based target tracking scheme is presented in [11]. The main distinction of our work is that it takes into consideration the discrete nature of the sampling intervals and extends the considerations of the importance of the timing within a particular epoch.

Among the works that are closer in spirit to our work are the prediction-based methods in [29, 30], both of which use the relative proximity to the current location and *predicted* target direction of travel for selecting the next tracking principal. However, these approaches do not specifically tie the predicted location of the moving target with the *next sampling* event, which is the main distinction of our work. Favoring the nodes that are closest to the current location of the tracked target as next tracking principals may be advantageous from the perspective of robustness in terms of not "missing" the coverage area of the subsequent principal. However, it imposes much higher overhead on the number of hand-offs between successive principals.

Another approach which exhibits a certain similarity with our work is presented in [31]. This approach also utilizes prediction of the moving object's location, however it differs from our methodology in the following aspects: (1) the focus in [31] is on reducing the duty-cycle of tracking nodes by promoting sleep-scheduling, without concerning the overall number of principals; (2) tracking principals are predetermined and selectively awaken for the arrival of the moving target, while, in our work they are determined on-the-fly; and (3) they do not take into consideration the sampling-interval information as a mean of reducing the number of tracking principals' transitions, although, concerns are expressed for the relationship between the sampling interval and the probability of loosing track of the target.

A very recent prediction-based tracking scheme which proposes a k-tracking nodes lookahead for the purpose of increasing the detection probability for fast-moving targets, at the expense of increased energy consumption, is presented in [1]. Our work, while not directly considering the impact of the high-speed motion still differs in the selection process of a leader – while we consider the overlap of the coverage with discrete (expected) location-points, [1] is based on the leaders proximity to the entire predicted trajectory.

A criteria similar to the maximal temporal coverage has been proposed in [23], however, our *implementation* of the criteria is different in the following three aspects: (1) we use the *next* predicted location as a basis for performing the evaluations, (2) it is designed for low computational overhead, a desideratum in sensor network applications with limited processing capabilities and (3) we consider the segment rooted at L_{k+1}, rather than the entire trajectory segment.

6. CONCLUSIONS AND FUTURE WORK

We addressed a specific aspect of the tracking problem in WSNs – minimizing the number of hand-offs among tracking principals – and proposed an improvement over the state of the art by considering the discrete nature of the location-sampling process. By carefully considering the concepts of effective and blind coverage, and their inter-dependency, we were able to generate a heuristic that incorporates them into the criterion for selecting the (expected) best candidate for the next-principal among the available sensor nodes in the 1-hop neighborhood of the current principal. Our experiments, in which we take into consideration variations of the effective sensing range, demonstrated that up to 25% (respectively, up to 31%) of the savings in the number of hand-offs can be achieved when compared to the RAB (respectively, to the CPS) approaches. In addition, the experimental observations demonstrated that our SLS approach can yield up to 30% savings when compared to RAB in denser WSN. Similar trends were observed when considering the variation of the objects' motions patterns.

There are several potential extensions to this work. In lieu of the results in [1] concerning fast moving object, it will be interesting to investigate the increase of the predicted time-instant for selecting the next tracking principal by considering nodes which are k-hops away from the current on. Our experiments have clearly showed that node density may represent a bottleneck in the achievable performances, yet increasing nodes densities may not be practical nor cost-effective. Even if the moving object's speed is low, this can enable us to increase the sampling interval to further reduce the number of tracking principals. Another factor that we

would like to consider in a more detailed manner is the impact of the accuracy of the location-estimation. A first step towards this would be to consider a probabilistic model of the quality of sensing and the impact of this (im)precision on the *pdf* of the expected locations of a given moving object.

An interesting practical consideration is to examine the impact of the mobility models which exhibit the "stop-and-go" behavior. Our long term goal is developing a larger-scale variant of our approach, so it can be applicable to tracking shapes that evolve over time like, for example, spreading of a wildfire via WSN. We also plan to explore incorporating techniques for probabilistic nearest-neighbor queries [25] into our tracking method.

7. REFERENCES

[1] A. Alaybeyogly, K. Erciyes, A. Kantarci, and O. Dagdeviren. Tracking fast moving targets in wireless sensor networks. *IETE*, 27(1):46–53, 2010.

[2] F. Bai, N. Sadagopan, and A. Helmy. IMPORTANT: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. In *IEEE INFOCOM*, pages 825–835, 2003.

[3] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *WICOMM*, 2(5):483–502, 2002.

[4] Q. Cao, T. Yan, J. Stankovic, and T. Abdelzaher. Analysis of target detection performance for wireless sensor networks. In *DCOSS*, pages 276–292, 2005.

[5] W.-P. Chen, J. C. Hou, and L. Sha. Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE TMC*, 3(3):258–271, 2004.

[6] S. Dodge, R. Weibel, and E. Forootan. Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. *URBAN*, 33(6):419–434, 2009.

[7] B. Gedik, K. Wu, P. S. Yu, and L. Liu. Processing moving queries over moving objects using motion-adaptive indexes. *IEEE TKDE*, 18(5):651–668, 2006.

[8] O. Ghica, G. Trajcevski, P. Scheuermann, Z. Bischoff, and N. Valtchanov. SIDnet-SWANS: A simulator and integrated development platform for sensor networks applications. In *SenSys*, pages 385–386, 2008.

[9] R. H. Guting and M. Schneider. *Moving Objects Databases*. Morgan Kaufmann, 2006.

[10] C. Hartung, R. Han, C. Seielstad, and S. Holbrook. FireWxNet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments. In *MobiSys*, pages 28–41, 2006.

[11] G. He and J. C. Hou. Tracking targets with quality inwireless sensor networks. In *ICNP*, pages 63–74, 2005.

[12] T. He, P. Vicaire, T. Yan, L. Luo, L. Gu, G. Zhou, R. Stoleru, Q. Cao, J. A. Stankovic, and T. F. Abdelzaher. Achieving real-time target tracking using wireless sensor networks. In *IEEE RTAS*, pages 37–48, 2006.

[13] S. Kim, S. Pakzad, D. E. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *IPSN*, pages 254–263, 2007.

[14] B. Liang and Z. J. Haas. Predictive distance-based mobility management for multidimensional pcs networks. *IEEE/ACM TON*, 11(5):718–732, 2003.

[15] Y. Liu, A. Choudhary, J. Zhou, and A. Khokhar. A scalable distributed stream mining system for highway traffic data. In *PKDD*, pages 309–321, 2006.

[16] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. TAG: a Tiny AGgregation service for ad hoc sensor network. In *OSDI*, pages 131 – 146, 2002.

[17] G. Mao and B. Fidan. *Localization Algorithms and Strategies for Wireless Sensor Networks*. IGI Global, 2009.

[18] M. Mokbel and W. Aref. SOLE: Scalable on-line execution of continuous queries on spatio-temporal data streams. *VLDB*, 17(5):971–985, 2008.

[19] S. Pattem, S. Poduri, and B. Krishnamachari. Energy-quality tradeoffs for target tracking in wireless sensor networks. In *IPSN*, pages 32–46, 2003.

[20] S. J. Prosser. Automotive sensors: past, present and future. *JPCS*, 76(1):1–6, 2007.

[21] J. Schiller and A. Voisard. *Location Based Services*. Morgan Kaufmann Publishers/Elsevier, 2004.

[22] R. Szewczyk, A. M. Mainwaring, J. Polastre, J. Anderson, and D. E. Culler. An analysis of a large scale habitat monitoring application. In *SenSys*, pages 214–226, 2004.

[23] E. Tanin, S. Chen, J. Tatemura, and W.-P. Hsiung. Monitoring moving objects using low frequency snapshots in sensor networks. In *MDM*, pages 25–32, 2008.

[24] G. Trajcevski, O. Ghica, and P. Scheuermann. Tracking-based trajectory data reduction in wireless sensor networks. In *SUTC/UMC*, pages 99–106, 2010.

[25] G. Trajcevski, R. Tamassia, H. Ding, P. Scheuermann, and I. F. Cruz. Continuous probabilistic nearest-neighbor queries for uncertain trajectories. In *EDBT*, pages 874–885, 2009.

[26] H. Wang, K. Yao, and D. Estrin. Information-theoretic approaches for sensor selection and placement for target localization and tracking in sensor networks. *JCN*, 7(4):438–449, 2005.

[27] Q. Wang, W. P. Chen, R. Zheng, K. Lee, and L. Sha. Acoustic target tracking using tiny wireless sensor devices. In *IPSN*, pages 642–657, 2003.

[28] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. D. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *SenSys*, pages 28–39, 2003.

[29] Y. Xu, J. Winter, and W. C. Lee. Dual prediction-based reporting for object tracking sensor networks. In *MobiQuitous*, pages 154–163, 2004.

[30] Y. Xu, J. Winter, and W.-C. Lee. Prediction-based strategies for energy saving in object tracking sensor networks. In *MDM*, pages 346–357, 2004.

[31] H. Yang and B. Sikdar. A protocol for tracking mobile targets using sensor networks. In *IEEE SNPA*, pages 71 – 81, 2003.

[32] Z. Zhong, T. Zhu, D. Wang, and T. He. Tracking with unreliable node sequences. In *INFOCOM*, pages 1215–1223, 2009.