Survey of mobile object tracking protocols in Wireless Sensor Networks: a network-centric perspective

Marjan Naderan, Mehdi Dehghan*, Hossein Pedram and Vesal Hakami

Computer Engineering and Information Technology Department,
Amirkabir University of Technology,
424 Hafez Avenue, Tehran, Iran
E-mail: naderan@aut.ac.ir
E-mail: dehghan@aut.ac.ir
E-mail: pedram@aut.ac.ir
E-mail: vhakami@aut.ac.ir
*Corresponding author

Abstract: Mobile object tracking in Wireless Sensor Networks (WSNs) has gained much attention during recent years owing to its growing application potential for the ubiquitous society. In this paper, we present an investigation and taxonomy of the state-of-the-art in tracking moving objects in WSNs. To this end, we define the target tracking problem in general and introduce the major design considerations for an efficient Object Tracking Sensor Network (OTSN) in particular. We typify a complete object tracking solution as a system of *signal processing-based* algorithms in charge of target identification/classification and *network-centric* protocols as the communication substrate. Our special highlight would be on the discussion of the network-centric protocols built around the notion of OTSN. We outline the advantages as well as the performance issues in relevant schemes and conduct a brief comparative study. The survey concludes with several key open problems together with a summary of directions for future works.

Keywords: wireless sensor networks; tracking; object; target; survey; taxonomy.

Reference to this paper should be made as follows: Naderan, M., Dehghan, M., Pedram, H. and Hakami, V. (2012) 'Survey of mobile object tracking protocols in Wireless Sensor Networks: a network-centric perspective', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. 11, No. 1, pp.34–63.

Biographical notes: Marjan Naderan received her BSc Degree in Computer Engineering in 2004 and the MSc Degree in Information Technology in 2006 both from Sharif University of Technology (SUT), Tehran, Iran. Since 2007, she is PhD student in Computer Engineering, with a major in computer networks, at Amirkabir University of Technology (AUT), Tehran, Iran. Her research interests include computer networks, wireless and mobile networks, object tracking, network optimisation and simulation of network protocols.

Mehdi Dehghan received his BSc Degree in Computer Engineering from Iran University of Science and Technology (IUST), Tehran, Iran in 1992, and his MSc and PhD Degrees from Amirkabir University of Technology (AUT), Tehran, Iran in 1995, and 2001, respectively. He joined the Computer Engineering and Information Technology (CEIT) Department of Amirkabir University of Technology in 2004. Currently, as an Associate Professor, he is the Director of the Mobile Ad hoc and Wireless Sensor Lab at AUT. His research interests include high speed networks, network management, mobile ad hoc networks and fault tolerant computing.

Hossein Pedram received his BSc Degree in Electrical Engineering in 1977 from Sharif University of Technology, Tehran, Iran. He continued with the MSc Degree in Electrical Engineering in 1980 from Ohio State University, Columbus, Ohio, USA and the PhD Degree in Computer Engineering in 1992 from Washington State University, Pullman, Washington, USA. He is currently an Associate Professor at the Computer Engineering and Information Technology (CEIT) Department in Amirkabir University of Technology (AUT), Tehran, Iran, and he is also the Director of the Asynchronous lab at AUT. His research interests include computer architecture, asynchronous design, network on chip, innovations in computer architecture, distributed systems and networking.

Vesal Hakami received his BSc Degree in Computer Engineering in 2004, and the MSc Degree in Information Technology in 2008 both from the Amirkabir University of Technology (AUT), Tehran, Iran. He is presently pursuing his PhD in Information Technology, at Amirkabir University of Technology, Tehran, Iran. His current research interests include context-aware and mobile computing middleware in ad-hoc mobile environments, self-stabilising protocols, quorum-based power saving schemes as well as target tracking and query processing in wireless sensor networks.

1 Introduction

Wireless Sensor Networks (WSNs) are typically comprised of a large number of battery-powered wirelessly connected sensor nodes with limited computational resources and capable of sensing one (or more) physical phenomena. A regular sensor node can be deemed as being made up of five basic units: a processor, a memory, a power supply, a radio transceiver, and one or more sensing components. Some WSNs may additionally rely on the functionality of being able to dynamically modify or control different components of a sensor subsystem. Nodes in these networks might feature an additional actuator part which is an electro-mechanical device used to actuate different sensing devices, adjust sensor parameters, move the sensor, or monitor power in the sensor node (Yick et al., 2008; Akyildiz et al., 2002).

Environmental *monitoring* and *tracking* of mobile objects have been marked as the two major categories of application areas for WSNs (Yick et al., 2008). In monitoring applications, usually a designated environment (indoor or outdoor) is supposed to be observed for a long duration, and upon detecting an event, the network needs to react by either sending an alarm to the user/application or performing an automatic operation via its actuators. Tracking mobile object(s), on the other hand, requires that the network chase the movement trajectories of the object(s) via activating nearby nodes and deactivating those far away from the target. The duration of a tracking mission is typically smaller than that of a monitoring application and the network may have to meet hard or real-time constraints as stipulated by the higher level entity (Yick et al., 2008). An indispensible subtask in a tracking system is the generation of periodic reports on the location of the target(s) over time and sending these reports towards the interested application/user. In both of the monitoring and tracking systems, it is deemed as a common practice to put the nodes not engaged in a mission into the sleep mode in order to reduce the energy overhead. Power conservation, in WSNs, typically leverages on scheduling mechanisms, prediction strategies, and/or signal processing-based algorithms.

Popular operating scenarios for monitoring and tracking applications in WSNs include: military target tracking and surveillance, natural disaster relief, tracking and surveillance of animals in wildlife preserves, biomedical health monitoring, tracking of humans in crowded and restricted areas, tracking of vehicles such as cars in highways, hazardous environment exploration and seismic sensing. In military target tracking and surveillance, a WSN can assist in intrusion detection and identification; among the specific example scenarios might be the spatially-correlated and coordinated troops and tank movements. In case of natural disasters, sensor nodes can sense and capture the environmental factors to provision for necessary data in order to forecast catastrophic phenomena before occurrence. In biomedical applications, surgical implants of sensors can help remotely monitor a patient's health. In seismic sensing scenarios, an ad hoc deployment of sensors across the volcanic area is leveraged to detect the development of earthquakes and eruptions.

Prior to the operational debut of the new generations of wireless distributed micro-sensor systems, the object tracking problem had been primarily investigated for robots and Personal Communication Services (PCS) networks (Bruce et al., 1997; Nebot et al., 2002; Parker, 1997; Lin and Hwang, 1996). Of pioneer studies in this area is the DARPA Distributed Sensor Network (DSN) program in the early 1980s (Chong et al., 2003) which, despite the original plans for the deployment of the project using a large number of tiny sensor nodes, was inevitably built over the traditional types of sensors (e.g., airborne radars) consistent with the technological limitations of the time.

During the last decade, object tracking in WSNs has been demonstrated to have promising application potential for the ubiquitous society, especially in Ubiquitous Sensor Networks (USNs) (Jin et al., 2006; Skibniewski and Jang, 2006; Kim et al., 2007). Efforts to implement real-life WSN surveillance and tracking systems have also been conducted by the research community, some of which are stated in Yick et al. (2008). The following projects are among the best-known demo deployments in the relevant literature:

- The *Line-In-The-Sand* project (Arora et al., 2004) features a sensor network of more than 90 nodes deployed to form a cooperative intrusion detection system with target classification and tracking capabilities. The project is, in essence, a technological synergy of signal processing-based algorithms, communications, networking and middleware services.
- The *ExScal* project (Arora et al., 2005) envisions a sensor network with more than 1000 sensor nodes along with 200 peer nodes wirelessly connected in an ad hoc fashion. The key Issues underlying this project include topology control, coverage, deployment and localisation.
- The *ZebraNet* project (Juang et al., 2002) deploys a mobile sensor network to track animal movements, especially a group of zebras. The number of zebras

under surveillance in this project was less than 20, and positional readings obtained from GPS are sent hop by hop across instrumented zebras to the base station.

- The *VigilNet* Project (He et al., 2006) characterises a sensor network of 40,000 nodes for surveillance missions to acquire and verify information about enemy capabilities and position of hostile targets. Several issues including localisation, time synchronisation, data aggregation, leader election and neighbour discovery have been investigated in this project.

Despite the fact that the emerging hardware technologies have facilitated the provisioning of many sensor-driven applications, when it comes to the design of an efficient object tracking solution, various special properties of a WSN still need to be taken into account. Some salient features of WSNs, which have both attracted and challenged the research community, include:

- providing a distributed milieu for computation
- automatic configuration of a large number of tiny sensor nodes
- limitation of resources, e.g., battery, memory and bandwidth
- error-prone communications, inevitable on account of the lossy nature of the wireless links and rapid node battery depletion
- low cost of deployment.

A WSN aimed for tracking targets is referred to as an Object Tracking Sensor Network (OTSN) (Xu et al., 2004).

In this paper, we investigate the latest trends in the design and deployment of OTSNs and present a classification for the various object tracking protocols proposed thus far in the relevant literature. Our objective is to provide a deeper understanding of the problem as well as to identify the important factors sought in an efficient mobile object tracking protocol for WSNs. We present a design-oriented list of criteria for evaluation and comparison of the prior art. In addition, we excerpt the fundamental commonalities inherent in existing proposals and introduce some of the open research issues yet to be addressed by the interested community. To the best of our knowledge, this is the first work undertaking an exhaustive survey in the area of mobile object tracking protocols in WSNs. Considering the broad range of topics that come into play when conducting research in the field of distributed tracking and recognising the growing interest in its potential application for many real-life scenarios, providing expository surveys on this area is even more encouraged.

The rest of the paper is organised as follows: in Section 2, we first define the general problem of object tracking and then go on with outlining the principal OTSN design factors. Section 3 is concerned with the taxonomic exploration of the target tracking WSN deployment scenarios as well as the brief description of the key modules at the core of a complete object tracking system. In Section 4, we moderately overview and highlight the strengths and weaknesses associated with some of the latest trends in devising network-centric protocols to function as the communications backbone of a tracking system. In Section 5, we come up with a brief comparative study of the prior art with reference to our given set of evaluation criteria. Section 6 is dedicated to the discussion of several open research problems together with the presentation of some promising directions for future work in this area. The survey concludes in Section 7.

2 Problem definition and design considerations

2.1 Problem definition

Problem scenario: An OTSN is assumed consisting of n wireless sensor nodes scattered throughout a certain geographical area for tracking m ($m \geq 1$) mobile objects. In general, nodes can be deployed across the region with no specific assumption made about their placement. Sensor nodes can detect the presence (or absence) of the targets by sampling the sensed signals from the environment. Each object enters and exits the region at a random time and at some random place, stays in each location with some probability p for some duration, moves independently from the other objects and disappears with probability $1 - p$. In general, we make no assumptions about the mobility models of the targets. A more detailed mathematical definition can be found in Oh et al. (2005).

Objectives: It is desirable to determine the locations of the targets. Through performing this operation periodically, one can keep track of the trajectories of the objects across the space-time continuum. Sensing nodes that detect the target(s) are supposed to send localisation reports towards the tracking application. Only a small number of nodes need to be active during each period such that the energy spent is minimised and the network lifetime is maximised. It is not desirable to lose the trajectories of the targets and in case of a loss, there must be some recovery mechanism in place with bounded error.

The region encompassing active nodes is referred to as the *monitoring region* (Xu et al., 2004; Zhang and Cao, 2004). Through utilising prediction methods, it is possible to preparatively activate nodes in regions that the targets are more likely to move on to, referred to as the *forwarding region* (Chen and Ann, 2005). In Cerpa et al. (2001), the tracking problem is viewed as forming a region spanning active sensors that besieges the target at any time, and is continuously maintained as the mobile object moves. Such a region is also termed as the *envelopment net* in Tsai et al. (2007), and its associated model is dubbed the *Frisbee* model (Cerpa et al., 2001).

The design of tracking protocols for WSNs is influenced by many challenging factors. We divide these factors into two groups which are elaborated in the following two subsections.

2.2 The interplay of tracking with WSN layers and services

Several assumptions on the functionality of other layers and services in WSNs have a significant impact on designing an object tracking solution and vice versa; in effect, many of the classical problems, such as power management, time synchronisation, medium access, neighbour discovery, data aggregation, group management, leader election and migration, need to be revisited so as to address the object tracking domain-specific requirements (Arora et al., 2005; He et al., 2006). A tracking method can either be implemented on top of the network layer (Tsai et al., 2007; Song and Hatzinakos, 2007; Chen et al., 2004; Tseng et al., 2004; Kulathumani et al., 2007), as it may need to be aware of the routing information, or alternatively it may be integrated as a partial component into the network layer itself (Zhang and Cao, 2004; Kung and Vlah, 2003; Liu et al., 2005; Tran and Yang, 2006; Demirbas and Lu, 2007). In Section 3, dedicated to the presentation of our taxonomy of OTSNs, we have differentiated between the methods specifically intended for integration into the network or transport layers (i.e., network-centirc tracking protocols) and those apt for functioning at other layers (e.g., signal processing-based algorithms). Here, we discuss the specifics of the interplay of tracking with the other layers, services and protocols, in line with the requirements presumable in OTSNs.

The scheduling algorithm: The sleep mode in sensor networks and its management through a scheduling algorithm can be considered as a mechanism to conserve energy and thus to increase network lifetime. Two different schemes have been widely adopted for utilising sleeping sensors in the literature (Fuemmeler and Veeravalli, 2008): nodes in the standby power-down mode can be re-activated by external means on an as-needed basis, for example as proffered in Xu et al. (2004), Yang and Sikdar (2003) and Brooks et al. (2003); or alternatively, the process might require to make modifications on the underlying MAC protocols as discussed in Gui and Mohapatra (2004) and Jiang et al. (2008). In Fuemmeler and Veeravalli (2008), the information on the object's trajectory is exploited to readjust the sleep-awake planning. However, as has been argued in Fuemmeler and Veeravalli (2008) and Pattem et al. (2003), at special times during which nodes are in sleep mode, the tracking error might increase. In Maheswararajah et al. (2009), the measurement error as well as the cost associated with using a particular sensor in detecting targets is the main factor in determining the specifics of the scheduling strategy.

MAC layer protocols: In general, medium access is more complicated than the scheduling algorithm which is only in charge of managing the on/off periods of the radio transceiver and/or the sensing unit of the nodes. Most of the existing tracking protocols, however, are agnostic towards the specifics of MAC layer design (as in Xu et al., 2007; Zhang and Cao, 2004; Tsai et al., 2007; Chen et al., 2004; Tseng et al., 2004; Kung and Vlah, 2003; Yang and Sikdar, 2003). In effect, while using cross layer information can greatly improve the performance of a tracking algorithm, it comes at the expense of designing a special-purpose MAC protocol. The work in Song and Hatzinakos (2007), specifically envisions a specialised MAC-level mechanism and clearly determines its interactions and message exchange patterns with the other layers. Important issues associated with the design of joint MAC layer and tracking protocols are still left as open problems.

Routing and querying protocols: An important factor in the design of routing protocols for specific use in the context of object tracking would be the possible existence of correlation between data owned by neighbouring nodes. If such a correlation exists, new routing metrics can be incorporated into decisions on path selection, e.g., in Liu et al. (2005) and Sung et al. (2007). However, in most of the existing tracking protocols, such correlation is not accounted for; instead, it has been assumed that either a routing algorithm is already available for sending reports towards the sink, as in Tsai et al. (2007), Song and Hatzinakos (2007), Chen et al. (2004) and Tseng et al. (2004), or alternatively as envisioned in protocols such as Xu et al. (2004), Yang and Sikdar (2003), Ji et al. (2004) and Chang et al. (2008), the information is supposed to be delivered to the sink simply via a single-hop connection to a top-level cluster head (or a tree root). Few works have explicitly envisaged specific routing mechanisms for steering tracking results back to the sink such as Zhang and Cao (2004), Kung and Vlah (2003), Liu et al. (2005), Tran and Yang (2006) and Liu et al. (2008). In fact, in these tree-based tracking protocols, the tree structure which is created during the first phase of the algorithm is utilised by the messages to retrace the route from the sink. In Hwang et al. (2008), a data reporting strategy is developed for object tracking which assumes the presence of mobile sinks; particularly, the sensor network is divided into grids and a weight function is used as a classification mechanism so as to determine whether an active node should send its report to the sink or not.

A similar trend can be noted in the context of querying mechanisms meant for object tracking in that it is also a lightly treated area. Only a few protocols such as GLANCE (Demirbas et al., 2006) and Distributed Quad Tree (DQT) (Demirbas and Lu, 2007) specifically deal with the problem of efficient data querying for tracking in WSNs by exploiting the geometry of the network. In these studies, a *distance sensitive* querying mechanism is presented which guarantees the cost of answering a query for an event to be at most a constant factor of the distance d to the nearest event in the network. Also, aside from the strategic use of an underlying hierarchical structure in tree-based schemes (Zhang and Cao, 2004; Kung and Vlah, 2003; Liu et al., 2005; Tran and Yang, 2006; Liu et al., 2008) – which is primarily intended to avoid the network-wide flooding of queries- in most of the relevant schemes, a tracking request is either assumed to be indiscriminately disseminated throughout the network or even no explicit mention of this problem has been made at all (Tsai et al., 2007; Song and Hatzinakos, 2007; Chen et al., 2004; Tseng et al., 2004; Yang and Sikdar, 2003).

Aggregation strategies: In-network aggregation is normally performed on data of the same type to reduce redundancy and to compress data with the prime intention of minimising energy consumption throughout the communication. A survey on aggregation strategies for WSNs can be found in Fasolo et al. (2007). In Liu et al. (2008) and Kumar et al. (2000), a tree-based structure for tracking mobile objects is proposed with special emphasis on data aggregation; it features a shortcut mechanism which augments the object tracking tree with some new edges so that the update and query costs are reduced. However, in most of the existing object tracking protocols, no specific aggregation strategy is prescribed or none of the existing approaches are explicitly incorporated into the final solution.

Data fusion techniques: Data fusion techniques are primarily utilised to draw inferences from the gathered data of sensor nodes as well as to reduce the error by eliminating noisy measurements (Smith and Singh, 2006; Hall and Llinas). Of closely related problems to data fusion is *Data Association* in which, measured data are associated to each target (track) with a degree of uncertainty (Liu et al., 2007; Smith and Singh, 2006). Data correlation is also cited as the other influential factor in devising fusion models (Smith and Singh, 2006). Various forms of data correlation are subject to a broad class of statistical techniques aiming at the identification of interrelations between random variables or data values. When it comes to WSNs, however, spatial and temporal correlations are assumed to be the models of prime interest (Vuran et al., 2004).

It has been demonstrated that the local fusion of captured data within sensor nodes enhances the quality of information and facilitates power savings in WSNs (Pashazadeh and Sharifi, 2007). Extensive studies have been conducted in the literature on data fusion techniques and a recent survey can be found in Smith and Singh (2006). More relevant proposals, such as Orguner and Gustafsson (2009), Fan et al. (2008), Pashazadeh and Sharifi (2007), have investigated the supplementary role of fusion techniques in target tracking systems. As for an example, the work in Fan et al. (2008) leverages on the LEACH algorithm (Heinzelman et al., 2002) for clustering the sensor nodes, and requires that each cluster head executes a fusion protocol consisting of two components: a one-step delay algorithm to select the tracks and a Fuzzy Cluster Means (FCM) algorithm to carry out data association tasks. In Pashazadeh and Sharifi (2007), a comparative study is conducted evaluating the accuracy of two voting algorithms for fusing target tracking data at the sink node. However, given that most of the fusion techniques heavily rely on signal processing algorithms, they are not further reviewed here to keep in tune with the communications-oriented theme of this paper.

Localisation service: A localisation service can be exercised at two stages throughout an object tracking mission: to estimate the whereabouts of the target based on the position of the detector nodes, or as well to determine the location of nodes in the initial phase of the network operation (Ali Alhmiedat and Yang, 2007). In almost all of the existing tracking protocols, it is assumed that sensors are aware of their positions and exchange this information with each other so as to locate their neighbouring nodes as well (Zhang and Cao, 2004; Tsai et al., 2007; Chen et al., 2004; Tseng et al., 2004; Kung and Vlah, 2003; Yang and Sikdar, 2003; Ji et al., 2004; Chang et al., 2008; Olule et al., 2007; Yu et al., 2004; Tran and Yang, 2006; Walchli et al., 2007; Wang et al., 2010). Tracking protocols also draw on localisation algorithms to either determine the location of the targets directly as elaborated in Tsai et al. (2007), Song and Hatzinakos (2007), Chen et al. (2004) and Tseng et al. (2004), or it might as well be simply assumed that the target location be approximated by the location of the nearest sensor node, much in the same manner as discussed in Zhang and Cao (2004), Kung and Vlah (2003), Wang et al. (2010) and Kulathumani et al. (2007). In Ali Alhmiedat and Yang (2007), the localisation techniques used in the context of tracking have been categorised into four groups: *prediction-based* such as Xu et al. (2004), Zhang and Cao (2004), Yang and Sikdar (2003), *sensing modality-based* as in Aslam et al. (2003), Collaborative Signal Processing (CSP)*-based* as envisaged in Li et al. (2002) and Chu et al. (2002), or the *ad-hoc* techniques discussed in Ali Alhmiedat and Yang (2007).

Node placement or the nodes location model: The fact that nodes are placed randomly or else deployed in a pre-planned fashion within the monitoring region is an important consideration in designing the object tracking algorithm. For instance, the work discussed in Tseng et al. (2004) is based on the assumption that nodes be deployed in a specific triangular fashion, or in Zhang and Cao (2004) and Kulathumani et al. (2007) nodes are placed in a grid structure. In Xu et al. (2004), Song and Hatzinakos (2007), Chen et al. (2004), Kung and Vlah (2003), Yang and Sikdar (2003), Tran and Yang (2006), Walchli et al. (2007) and Wang et al. (2010), however, no specific placement model has been envisaged.

Mobility model or the movement prediction model of the target: If the movement model or the velocity of the target is known by the nodes in advance, or at least the movement trajectories of the targets can be predicted, as in Xu et al. (2004), Yang and Sikdar (2003), Tseng and Lu (2009), Goel and Imielinski (2001), targets are less likely to be missed. Nevertheless, it is often more desirable not to rely on the existence of a priori information on the objects' movements. It should also be noted that not every mobility model can be applied to every environment. For example, the work presented in Kung and Vlah (2003) draws on the assumption of the locality of movements; i.e., the targets are assumed to usually move within a small restricted region, say, for instance, people walking or strolling along the streets of a city.

2.3 Criteria related to quality of tracking

In this subsection, we present the most frequently referenced set of criteria for assessing the quality of tracking in sensor networks, for instance in the form of simulation measures as in Xu et al. (2004), Zhang and Cao (2004), Song and Hatzinakos (2007), Chen et al. (2004), Kung and Vlah (2003), Yang and Sikdar (2003), Gui and Mohapatra (2004). Some of these metrics are also well established in the more general 'quality of service' parlance, and we elaborate, here, on their subtleties from the object tracking perspective.

Scalability: The tracking protocol should be designed with the scalability property in mind so that it could account for tracking multiple targets and for increasing number of query requests targeted to different regions, to different objects or issued for more accuracy; the protocol should also scale as the monitoring region under its coverage expands. In contrast to methods relying on centralised algorithms such as Kung and Vlah (2003) and Lee et al. (2006), tracking protocols with a cluster-based structure have been shown to be scalable against both a large number of sensor nodes as well as in scenarios involving more than one target (Chen et al., 2004; Yang and Sikdar, 2003; Wang et al., 2010). Algorithms capable of tracking multiple targets are referred to as Multi-Target Tracking or MTT schemes. In some single target tracking protocols, however, several properties must change if the MTT functionality is desired; e.g., as outlined in Song and Hatzinakos (2007) and Li et al. (2002). This issue is further discussed in Section 3. Tracking multiple objects simultaneously is closely related to the data association problem referred to in Section 2.2, under 'Data fusion techniques'.

Tracking precision: In the simplest sense, how far the estimated location of each target, as derived from samples and localisation algorithm, differs from its real location is interpreted as the tracking precision. Since target localisation based on a larger number of nodes usually requires more computational effort, one can trade tracking precision for computational complexity. Such correlation also exists between tracking precision and: probability of target loss as discussed in Xu et al. (2004), Yang and Sikdar (2003), Tseng and Lu (2005), recovery mechanisms for relocating the object in Yang and Sikdar (2003), prediction accuracy as in Xu et al. (2004), and sensitivity to noisy measurements (Chen et al., 2004) as well as the ratio of false alarms.

Tracking delay: This metric plays a key factor in two stages of the tracking operation:

- the time it takes to estimate the location of a given target from the very moment it appears
- the latency of reporting the estimated location to the sink.

The delay parameter associated with the second phase is heavily dependent on the specifics of the routing or the data dissemination protocol (as stated in Chen et al., 2004). Despite no special measures have been taken for explicitly guaranteeing delay constraints in the majority of the existing object tracking protocols (e.g., Xu et al., 2004; Zhang and Cao, 2004; Tsai et al., 2004; Song and Hatzinakos, 2007; Chen et al., 2004; Kung and Vlah, 2003; Yang and Sikdar, 2003), some recent works have been proposed which instead factor the *quality of service* parameters such as delay into the underlying routing protocol in WSNs (He et al., 2003; Felemban et al., 2006; Huang and Fang, 2007).

Energy consumption: The lifetime of the network is inversely proportional to the energy overhead it incurs. It can be easily argued that from a designer's perspective, power conservation and quality of tracking are of two conflicting interests; thus, there must be some kind of a tradeoff between the desired quality criteria and energy usage.

Adaptability: Whether the tracking algorithm depends on a specific target mobility model or is able to track high speed targets while incurring little performance degradation, is an important factor in evaluating target tracking systems. For instance, in Kung and Vlah (2006) it is assumed that the targets' trajectories are based on the *locality of movements*, while in Tsai et al. (2007), the protocol encompasses a *face track shortening* phase for tracking high speed targets.

Degree of coverage: This metric indicates what proportion of the monitoring region is covered by nodes with a given sensing range to track the target efficiently. For example, as shown in Gui and Mohapatra (2004), it is possible to reach acceptable Quality of Surveillance (QoSv) with a partial coverage of the monitoring region. Several other existing works inspecting this aspect of target tracking have been proposed in Yang and Sikdar (2003), Gui and Mohapatra (2004), Pattem et al. (2003), Chakrabarty et al. (2002) and Wang et al. (2008).

Protocol overhead: This entails the control packets overhead and the pre-processing cost of the algorithm. Usually, protocols that rely on a specific structure (e.g., Zhang and Cao, 2004; Tsai et al., 2007; Chen et al., 2004; Tseng et al., 2004; Kung and Vlah, 2003) need more control packets compared to unstructured protocols (e.g., Song and Hatzinakos, 2007; Chu et al., 2002) which are normally spared from the initial cost of the structure setup and/or its maintenance over time.

Table 1 gives a summary of all the design considerations and evaluation metrics discussed in this section and portrays the interrelation between factors affecting the algorithm design and their corresponding quality of tracking metrics. As can be seen in the table, the specifics of the protocols and services in different layers are tightly coupled with several different quality of tracking parameters.

For instance, while relying on a priori information on the mobility model of the target(s) may limit the scalability or even adversely affect the adaptability of the tracking algorithm, it might instead help decrease the energy consumption, which is evidently a desirable property.

Table 1 The interrelation between design characteristics and quality of tracking metrics

Layers, services, and protocols	*Quality of tracking metrics*
Scheduling mechanism	• Scalability • Tracking precision • Degree of coverage • Nodes energy consumption • Control packets overhead
MAC layer protocols	• Scalability • Nodes energy consumption • Control packets overhead
Routing, querying, aggregation and data fusion strategies	• Scalability • Tracking precision • Tracking delay • Nodes energy consumption • Control packets overhead
Localisation service	• Tracking precision • Nodes energy consumption • Control packets overhead
Node placement or location model	• Tracking precision • Nodes energy consumption • Degree of coverage • Control packets overhead
Mobility model of the target	• Scalability • Nodes energy consumption • Adaptability

3 Towards a taxonomy of OTSN: deployment scenarios and classification of tracking techniques

In this section, we first distinguish between the possible deployments of mobile OTSNs, and describe their main characteristics together with the associated issues and requirements. We then present a typification of a complete target tracking solution and go on to introduce our classification of the most representative categories of the prior art in network-centric object tracking protocols for WSNs.

3.1 OTSN deployment scenarios and applications characterisation

A straightforward configuration-oriented taxonomy of the possible OTSN deployment scenarios can be based on the following three dimensions:

1 The number of the objects to be tracked
2 The type of the objects to be tracked
3 The modality of the sensors.

In what follows, we further discuss the implications a particular type of scenario might bring on the design of a tracking system.

3.1.1 The number of objects

Depending on the number of objects the OTSN is supposed to track, two obvious application scenarios can be envisaged:

- single target tracking
- Multi-Target Tracking (MTT).

The methods presented in Xu et al. (2004), Tsai et al. (2007), Chen et al. (2004), Tseng et al. (2004), Kung and Vlah (2003), Yang and Sikdar (2003), Xu et al. (2004), Lin et al. (2006), Olule et al. (2007), Yu et al. (2004), Tran and Yang (2006) and Lee et al. (2006) have the ability of tracking multiple targets, while the algorithms discussed in Zhang and Cao (2004), Song and Hatzinakos (2007), Goel and Imielinski (2001), Chu et al. (2002), Walchli et al. (2007), Wang et al. (2010) and Kulathumani et al. (2007) are able to track only a single object.

For some tracking protocols, several considerations must be made in case the MTT functionality is desired. For instance, as stated in Song and Hatzinakos (2007), to support tracking of multiple targets would require that orthogonal channels be utilised using techniques such as TDMA or FDMA. In Chen et al. (2004), despite the tracking algorithm is primarily proposed for single target tracking, provisions can be made in the protocol, such as the introduction of special *signature* packets,[1] so that it can be extended to support tracking multiple targets as well. In Liu et al. (2003), a group formation algorithm is proposed which extends the method in Chu et al. (2002) to track multiple targets. In this work, a dynamic group management method is presented to initiate and maintain multiple tracks in a distributed manner and each group is responsible for tracking a single target. As briefly explained in Section 3.2, MTT support might also call for reconsiderations on the specifics of the signal processing-based algorithms (e.g., data association techniques) used in a tracking system.

3.1.2 The type of objects

Depending on the type of object, tracking scenarios can be categorised into two groups:

- discrete object tracking
- continuous object tracking.

Discrete (individual) objects are ordinary targets like humans, cars, animals and so forth which may emit noise, light, and seismic waves, can be separated from each other and are countable. A continuous object, on the other hand, might constantly extend across the region and typically occupies a large area (Ji et al., 2004; Chang et al., 2008). They tend to diffuse, expand in size, change in shape or even split into multiple relatively smaller continuous objects. Of typical examples might be diffused gas, biomedical and chemical liquid. The main challenge associated with the tracking of this type of targets is the non-trivial fusion and/or dissemination of local boundary information which must be performed in a dynamically adaptable fashion. The works presented in Ji et al. (2004) and Chang et al. (2008) specifically address the tracking of continuous objects.

OTSN deployment may also be particularly intended for tracking objects of specific mobility model, speed range, or nature:

- The mobility model/type of the objects:
  - city mobility model (Kung and Vlah, 2003; Lin et al., 2006)
  - random mobility models; e.g., random waypoint (Yen and Yang, 2006)
  - associative or group mobility models (Ochirsuren et al., 2008).
- The speed range of the objects:
  - *low speed*; e.g., pedestrian walking (Kung and Vlah, 2003), speeds up to 15 m/s (Yang and Sikdar, 2003), or slow movement scenarios with 0~10 m/s velocities (Tsai et al., 2007)
  - *moderate speed* with 10~20 m/s velocities as envisaged in Tsai et al. (2007)
  - *high speed*; e.g., cars in highways, speeds up to 400 km/h (Walchli et al., 2007), fairly fast scenarios with velocities up to 20 m/s (Zhang and Cao, 2004) and 30 m/s (Xu et al., 2004), or the 20~30 m/s velocities featured in Tsai et al. (2007).
- The nature of the objects:
  - *intruder/adversary*; e.g., on-demand patrol in Gui and Mohapatra (2005) and foreign objects in Stark and Davis (2004)
  - *friendly and collaborating*; e.g., coverage-oriented patrol in Gui and Mohapatra (2005) and friendly objects in Stark and Davis (2004).

However, as pointed out earlier in Section 2, most tracking methods are not tightly coupled with mobility characteristics of the objects, essentially favouring generality over efficient but tailored 'case' studies.

3.1.3 The sensing modality

Although many of the existing tracking protocols (e.g., Xu et al., 2004; Zhang and Cao, 2004; Tsai et al., 2007; Tseng et al., 2004; Kung and Vlah, 2003; Yang and Sikdar, 2003; Goel and Imielinski, 2001; Xu et al., 2004; Lin et al., 2006; Olule et al., 2007; Tran and Yang, 2006; Lee et al., 2006; Kulathumani et al., 2007) do not explicitly envision a particular type of sensor, the modality of sensors used for tracking targets might influence the specifics of the signal processing protocols, the localisation or estimation algorithm as well as the type of objects to be tracked; for instance, the set of equations used for modelling acoustic waves in Chen et al. (2004), may not apply to the processing of other types of signals.

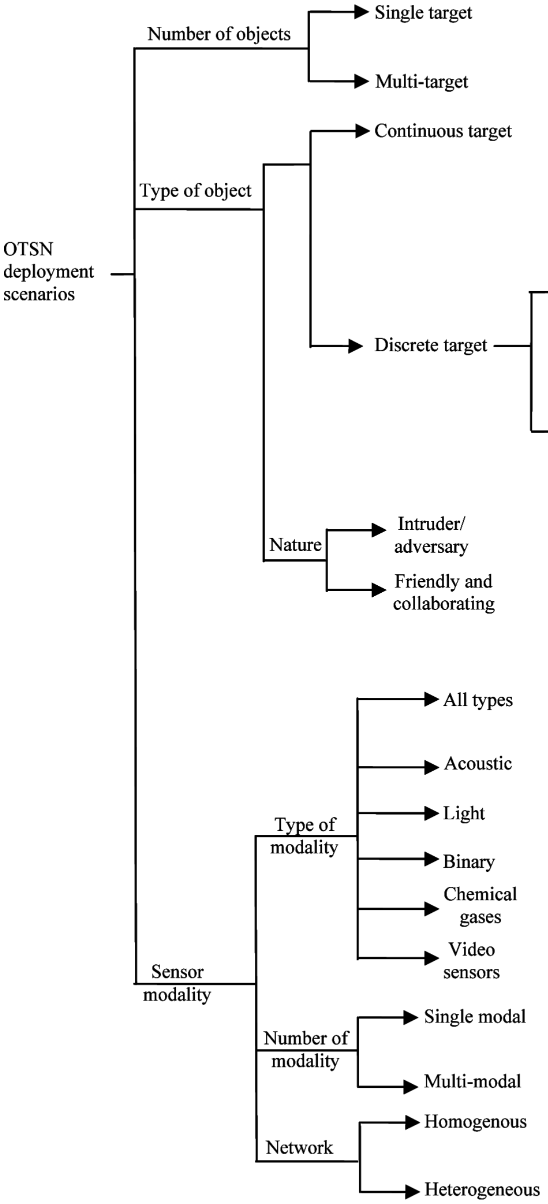
The methods discussed in Song and Hatzinakos (2007), Chen et al. (2004), Chu et al. (2002), Yu et al. (2004) and Wang et al. (2010) explicitly assume that the network consist of acoustic sensors. The works in Ji et al. (2004) and Chang et al. (2008), which are intended for continuous object detection, assume that sensors are capable of perceiving chemical gasses. The tracking scenario in Walchli et al. (2007) relies on light sensors to detect a moving person holding a flashlight. Algorithms presented in Aslam et al. (2003), Kim et al. (2005), Shirvastava et al. (2006), Mechitov et al. (2003), Aghajarian and Berangi (2008) draw on *binary sensors* to fulfil tracking missions. A binary sensor produces only one bit of information indicating whether the object is getting closer to the node or moving farther away; typically, these sensors do not measure the distance from the object.

A broad array of algorithms also exist which are intended for tracking scenarios in the domain of visual sensor networks, some of which can be found in Gurses et al. (2009), Monari et al. (2008), Pahalawatta et al. (2004), Soro and Heinzelman (2005) and Zhang et al. (2006). In these scenarios, sensors are essentially video cameras with a limited directional Field of View (FOV), and it is desirable to track the trajectories of object(s) in the form of a sequence of video frames. Further details regarding the visual modality are closely related to the topic of *image processing* and lie outside the scope of this paper.

A more refined differentiation of OTSN deployments w.r.t. sensing modality can be drawn on both individual node and network levels; i.e., *single-modal* vs. *multi-modal* nodes and accordingly, *homogenous* networks (where nodes feature the same sensor type) vs. *heterogeneous* networks (in which different nodes may have different types of sensors). For instance, it is assumed in Huiyu et al. (2008) that the WSN consists of multi-modal sensor nodes featuring a camera mounted between two microphones. Examples of target tracking in a heterogeneous sensor network can be found in Kushwaha et al. (2008) and Amundson et al. (2007) wherein some nodes are acoustic and others are equipped with a camera. A general discussion on exploiting heterogeneity in WSNs can be found in Yarvis et al. (2005); however, as argued in Abidi et al. (2008), heterogeneous target tracking is more common in wide area surveillance systems.

Figure 1 summarises our categorisation of the OTSN deployment scenarios.

Figure 1 OTSN deployment scenarios



3.2 Classification of mobile object tracking techniques in WSNs

Few works have recently investigated the state of research in WSN target tracking. However, given the ongoing nature of the problem, complexity and the non-obvious taxonomy of the available schemes, a solid classification is not yet established. A five-group categorisation of target tracking methods for sensor networks has recently been proposed in Bhatti and Xu (2009), namely: hierarchical, tree-based, prediction-based, Mobicast message-based and hybrid.

As discussed in Bhatti and Xu (2009), the hierarchical category identifies those tracking protocols featuring a clustering procedure to create a hierarchical control structure for the network. In tree-based schemes, the underlying communications backbone takes the form of a tree structure. Prediction-based approaches utilise a prediction strategy to calculate the next location of the target. The mobicast message-based category represents special multicast protocols which are leveraged to wake up the most pertinent subset of nodes preparatory to the target movement. As the name suggests, *hybrid* methods combine elements from any pair of the aforementioned approaches.

It can be argued that the classification scheme presented in Bhatti and Xu (2009) is not specific enough, settles on a rather coarse categorisation and to resolve the ambiguity in characterisation of the exiting methods demands a more precise and finer distinction between the groups. For instance and as is discussed in the next section, despite the assumptions made in Xu et al. (2004a, 2004b) are also compatible with the definitions given for hierarchical and hybrid structures, these two approaches have been classified under the ‘prediction-based’ schemes in Bhatti and Xu (2009).

Additionally, Tsai et al. (2007) have proposed a primitive two-group classification of the WSN tracking schemes: *cluster*-based and *non-cluster*-based. It is argued in Tsai et al. (2007) that as a common ground in tracking solutions, signal processing-based algorithms or alternatively prediction-based schemes are utilised for reducing the energy consumption Xu et al. (2004), Yang and Sikdar (2003), Goel and Imielinski (2001) and Xu et al. (2004). Mobicast message exchange protocols are also referred to in Tsai et al. (2007) as a mechanism to activate sensor nodes across the monitoring region in preparation for the target's future movement.

In this paper, we characterise a complete object tracking solution as a hybridised system of two major groups of algorithms and protocols:

- signal processing-based algorithms
- network-centric protocols.

Within the perspective of signal processing-based algorithms, the topological structure of the network is typically avoided as an issue, and in most of the existing methods, it is simply assumed that sensors would be grouped in to clusters within which only one node is given the charge of gathering data and tracking the target; for instance, as envisaged in Chong et al. (2003), Oh et al. (2005), Li et al. (2002), Liu et al. (2007), Shin et al. (2003) and Brooks et al. (2003). These algorithms help reduce energy consumption through the intelligent activation of only the most pertinent subset of nodes and leaving others in sleep mode (Tsai et al., 2007).

From the viewpoint of the WSN layered communication model, signal processing-based algorithms would best serve at the application layer, intended mainly for the estimation of the next state of the target given the previous history and based on current measurements. The specifics of the underlying communications mechanism, such as routing or scheduling, would be kept transparent from the CSP layer; for instance, the information-driven target tracking scheme discussed in Zhao et al. (2002) has envisaged the black-box incorporation of the existing WSN routing protocols such as *Directed Diffusion* (Intanagonwiwat et al., 2003) as its underlay networking service (Shin et al., 2003; Chu et al., 2002).

A classification of CSP-based algorithms, used in the context of target tracking, is presented in Liu et al. (2007) on the basis of the number of targets; i.e., *single* vs. *multi* target tracking. Single target tracking methods, depending on their data estimation technique, are further sub-classified into: *sequential Bayesian estimation* (Liu et al., 2007), *Kalman filter* (Liu et al., 2007) and *particle filter*-based (Doucet et al., 2001) methods. CSP-based MTT protocols, on the other hand, might in turn leverage on data classification techniques (Li et al., 2002), data association processing (Liu et al., 2007) as well as identity management (Oh et al., 2005; Shin et al., 2003). The major categories of data association techniques identified in Liu et al. (2007) are Multiple Hypothesis Tracking (MHT) Liu et al. (2007), Joint Probabilistic Data Association Filter (JPDAF) Liu et al. (2007), or more recent emerging trends such as *Monte Carlo-based sampling* (Oh et al., 2004) and *graphical models* (Chen et al., 2005).

However, a comprehensive taxonomy of CSP-based methods is yet to be established so as to account for the wide assortment of algorithms built around the notion of object tracking in WSNs. As the study of these algorithms stays outside the scope of this paper, henceforth, we skip discussing any further particulars of this category.

Network-centric protocols, on the other hand, deal mostly with the communication-oriented concerns such as gathering data and reporting the location of the targets. These protocols may preferably be implemented within the network or the transport layers of the communication architecture. A moderate overview of the best-known schemes in this category is given in Section 4.

Figure 2 demonstrates the correspondence between the five-layer WSN protocol stack (Akyildiz et al., 2002) and the protocols used in an object tracking solution. The heavily outlined box on the right shows the areas of emphasis in this survey.

Figure 2 Correspondence between an object tracking system and WSN protocol stack, with our focus in this paper as the bold box

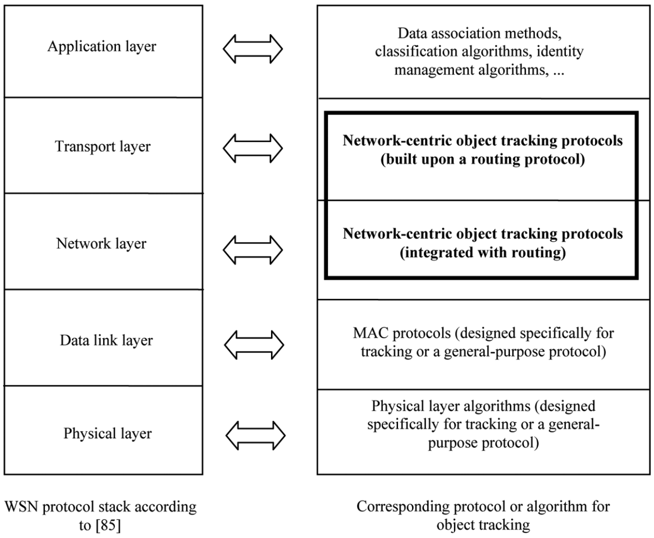


Figure 3 illustrates our classification of network-centric object tracking protocols. From the *structural* perspective, network-centric protocols can be grouped into three main categories:

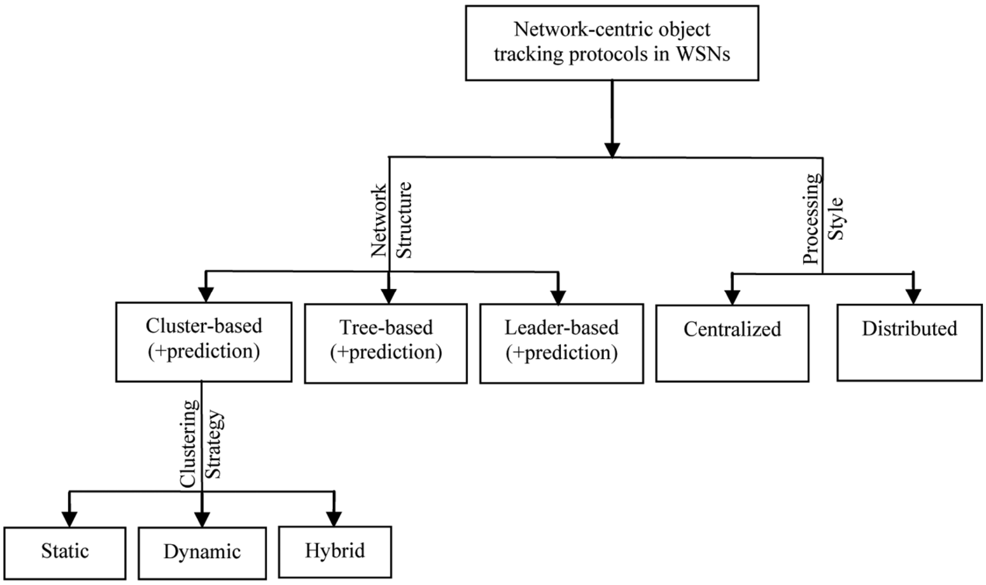
- cluster-based
- tree-based
- leader-based.

Additionally, the protocols within each category can optionally be augmented with prediction strategies which help reduce both the energy consumption as well as the probability of target loss.

Clustering can be performed offline, i.e., before the network takes on a tracking mission (Xu et al., 2004;

Tsai et al., 2007; Chen et al., 2004; Yang and Sikdar, 2003; Goel and Imielinski, 2001; Olule et al., 2007; Yu et al., 2004), or online, i.e., upon detection of a target (Ji et al., 2004; Chang et al., 2008; Wang et al., 2010). Typically, a single node, namely the *cluster head* would be determined to be in charge of managing each cluster Nodes detecting target(s) send their sensed data towards the heads. Once sufficient information gathers up in the cluster head, it would estimate the location of the target and report to the sink. In the event that a target moves out of the range of the current cluster, a new cluster head takes over, nodes in the previous cluster switch to sleep mode and the process is repeated.

Figure 3 Network-centric object tracking protocols in WSNs: a taxonomy



Depending on the clustering strategy adopted, cluster-based techniques can be further sub-classified into:

- static clustering
- dynamic clustering
- hybrid schemes.

In static clustering, nodes located within the monitoring region are grouped into clusters on the basis of some conventional measure (e.g., remaining energy of the nodes (Heinzelman et al., 2002), physical distance as in the Voronoi diagram-based method in Chen et al. (2004), etc.), and the cluster structure is not updated in reaction to the targets' movements. Dynamic clustering, on the other hand, builds clusters of nodes upon the target's entrance to the monitoring region and in an online fashion. The key issue in tracking protocols with dynamic cluster formation is the detection of boundary nodes, addressed only in Ji et al. (2004), Chang et al. (2008), Park (2006) and Wang et al. (2010). Evidently, the adoption of a dynamic strategy for clustering enhances the accuracy of tracking, albeit at the cost of increasing the overhead (Wang et al., 2010). Finally, the hybrid schemes attempt to strike a balance between the accuracy achieved through well-defined clusters around the target on the one hand and control overhead on the other by applying a semi-dynamic cluster formation strategy. For instance, the HCTT protocol (Wang et al., 2010) leverages on dynamic clusters over an already established static structure to track the target movements between neighbouring clusters or on a cluster boundary. Whether a hybrid scheme would be an efficient alternative in each case – i.e., in terms of target miss ratio and energy consumption – depends on the actual requirements of the application and its usability should be weighed against the overheads involved.

In tree-based methods, a tree structure is set up amongst nodes online, i.e., upon detection of the target (Zhang and Cao, 2004; Kung and Vlah, 2003; Lin et al., 2006; Tran and Yang, 2006; Lee et al., 2006). Necessary message exchanges take place so that each node determines its parent and accordingly remains in (or be removed from) the tree structure. Each node detecting the target passes its information to its parent to be ultimately delivered to the root of the tree. The root is in turn responsible for sending reports to the sink.

Leader-based methods are analogous to clustering protocols with the only difference that no specific structure is set up amongst nodes and only those detecting the target participate in the tracking process (Song and Hatzinakos, 2007; Tseng et al., 2004; Chu et al., 2002; Walchli et al., 2007). To this end, a node (usually the one nearest to the target) is elected as the leader which, after gathering enough information from its subordinates, would estimate the target's location and report to the sink.

Network-centric object tracking protocols can also be classified according to their style of processing into: *centralised* vs. *distributed*.

In centralised schemes, a designated entity is assumed to get hold of the knowledge of the whole network, obtained presumably by requiring that all nodes send on the necessary information, and build the desired structure for tracking. For instance, the tree construction algorithm proposed in Kung and Vlah (2003) operates in a centralised fashion with the knowledge of the weights of all the links. In Lee et al. (2006), the problem of building a minimum cost object tracking tree is formulated in terms of a '0–1 integer programming' model for which a Lagrangean Relaxation-based (LR-based) heuristic algorithm is proposed. In Yu et al. (2004), a sensor/server model is presented in which sensor nodes send their information to the server and the server maintains a list of all nodes along with their corresponding states. A similar centralised processing style has also been undertaken in Xu et al. (2004), Goel and Imielinski (2001), Xu et al. (2004) and Olule et al. (2007).

In contrast to centralised methods, in distributed tracking protocols, neighbouring nodes cooperate with each other to set up the communication structure, be it a cluster or a tree, for interaction with the sink node (Zhang and Cao, 2004; Tsai et al., 2007; Song and Hatzinakos, 2007; Chen et al., 2004; Tseng et al., 2004; Yang and Sikdar, 2003; Chu et al., 2002; Lin et al., 2006; Tran and Yang, 2006; Walchli et al., 2007; Wang et al., 2010).

Table 2 lists some of the latest tracking methods proposed in the literature that can be framed under this classification.

Table 2 Classification of the network-centric object tracking methods

Structure type/ strategy	*Method*
Cluster-based	• Dynamic clustering method for acoustic sensors (Chen et al., 2004) • DOT (Tsai et al., 2007) • DCS (Ji et al., 2004) and its optimisation CODA (Chang et al., 2008) • RARE (Olule et al., 2007) • HCTT (Wang et al., 2010)
Tree-based	• DCTC (Zhang and Cao, 2004) and its optimisation in Zhang and Cao (2004) • DAB (Kung and Vlah, 2003) and its optimisations DAT and Z-DAT in Lin et al. (2006) • OCO (Tran and Yang, 2006) • Heuristic method (Lee et al., 2006) • Trail (Kulathumani et al., 2007)
Leader-based	• LESOP architecture (Song and Hatzinakos, 2007) • Mobile agent based method (Tseng et al., 2004) • IDSQ (Chu et al., 2002) • DELTA (Walchli et al., 2007)
Cluster-prediction	• PES (Xu et al., 2004) and DRP (Xu et al., 2004) • DPT (Yang and Sikdar, 2003) • PREMON (Goel and Imielinski, 2001) • Adaptive tracking (Yu et al., 2004)
Tree-prediction	• DCTC (Zhang and Cao, 2004) (prediction-based scheme)

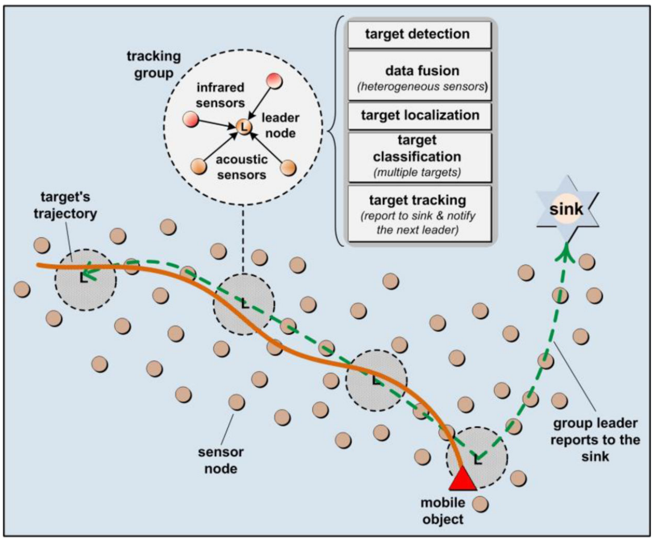
4 Mobile object tracking protocols in WSNs: an overview

In this section, we moderately discuss some of the latest WSN target tracking schemes that promote a network-centred mentality. We organise our discussion under three principal headings, viz., leader-based, tree-based and cluster-based methods, in line with the classification outlined in the previous section.

4.1 Leader-based tracking methods

As discussed earlier, leader-based methods typically feature a lightweight structure-less protocol and require that only detecting nodes participate in the tracking mission. Figure 4 depicts a simple leader-based tracking network in which only nodes near the object are active. Despite their simplicity, these protocols are primarily intended for single target tracking applications and to cater for multi-target scenarios requires inevitably complex modifications. Four leader-based tracking protocols, namely: LESOP, IDSQ, DELTA, and a mobile agent approach are discussed in this subsection. A fundamental commonality in all these protocols is that no predefined structure is relied upon and that in every instance of time only one node takes charge of tracking the target.

Figure 4 A simple leader-based target tracking network: detecting nodes send their information to an elected leader node which subsequently delivers the object's information to the sink (see online version for colours)



4.1.1 Low Energy Self-Organising Protocol (LESOP)

The LESOP method (Song and Hatzinakos, 2007) fosters a two-layered architecture, namely *Embedded Wireless Interconnection* (*EWI*) which enables cross layer interaction between the application and MAC layers. The object tracking functionality at the upper layer draws on the communication and sensing primitives provisioned for via specific MAC and physical layer protocols integrated within the lower levels of the EWI architecture.

In LESOP, the node first detecting the target is elected leader which prompts its neighbours to perform sensing and also chooses the node with the highest locally calculated detection fusion coefficient to be the surrogate leader. Only a subset of the neighbouring nodes participate in the fusion process via sending their detection information to the surrogate. The number of nodes within this subset is determined in reconciliation with the optimum gain on the precision of the target's location; i.e., the minimum improvement ratio of accuracy at the cost of increasing the number of detecting nodes by one. The surrogate leader builds a profile of the target's trajectory on the basis of the linear combination of the locations of detecting nodes together with the track information it receives from the primary leader. Once a predefined time period elapses, the surrogate takes the role of the primary leader and the procedure repeats.

Sensor nodes in LESOP are assumed to be equipped with two radios: a primary RF radio and a secondary wakeup radio. The primary radio is used for routine wireless data packet transmission. Its transmitting power and rate are assumed to be fixed. This radio can operate in one of the three distinct modes: transmitting, receiving/idle, or sleeping. The secondary wakeup radio can only send or detect busy tones. The power consumption of wakeup radio is of the same level as the sleeping mode of primary radio, which is negligible. Furthermore, it is assumed that the wakeup radio has the capability of sending/detecting busy tones at two separate frequencies.

In sum, LESOP consists of a lightweight application-layer tracking protocol together with relatively complex MAC and physical layers. It, however, provides only for transmissions within a single hop and does not explicitly account for multi-hop connections. This can be attributed to the fact that no specific network/transport layer protocol is envisioned within the system architecture. In addition, to track more than one target, not only multiple orthogonal radio channels, (e.g., through the use of TDMA or FDMA), are needed but it may also warrant modifications in both MAC and application layers. The track information hand-over from a previous to a new leader also increases the latency, while this information might not even be used for prediction of future target movements.

4.1.2 A mobile agent-based approach to target tracking

In Tseng et al. (2004), sensor nodes are assumed to be arranged in a triangular structure. However, provisions can be made in the protocol to serve in networks of irregular shape as well; for instance, through the exploitation of *Voronoi* diagram (Aurenhammer, 1991). It is also assumed that every node knows both its own location as well as the locations of its neighbours, and that it is also capable of determining its distance to the target.

Upon detecting an object, an election process (Tseng et al., 2004) is conducted by nearby sensors to choose a node on which a *master* agent is initiated. As the object moves, the agent may also migrate to a closer sensor to maintain its monitoring status. The *master*, through dispatching *slave* agents, invites exactly two of its neighbouring nodes in order to assist in the tracking mission. These three agents cooperate to perform the *Trilateration* algorithm. Slave nodes periodically report their sensing results to the master who is supposed to calculate the object's location. As the object moves, the slave agents may be revoked and reassigned in concert with certain thresholds of signal strength. To reduce the sensing overhead, the tracking agents prevent other irrelevant sensors from monitoring the object by periodically sending warning signals.

In order to extend the protocol to cater for irregular deployment of sensor nodes, the election process does not need to be changed but the rules to migrate masters/slaves need to be modified and sensors need to know the locations of at least their two-hop neighbours. To find a master and two slave agents in an irregular shape network, the classical Voronoi graph problem can be exploited (see Tseng et al., 2004 for details).

As envisioned in Tseng et al. (2004), the specifics of the fusion and delivery of the tracking results can be strategised in either of the Non-Agent-Based (NAB), Threshold-Based (TB) or Distance-Based (DB) modes. NAB, in effect, serves as a referential strategy for comparison. It assumes that each sensor operates independently and forwards its sensing results back to the gateway periodically. In TB, the master agent accumulates the tracking results and carries them along until their volume exceeds a predefined value, in which case, the results are delivered to the gateway through the shortest path available. In DB, the delivery action may be taken only when the master agent moves.

Given its reliance on an underlying routing algorithm, the tracking protocol in Tseng et al. (2004) should operate at the application (or more specifically, on top of the network) layer. Compared to LESOP, it does not rely on specific MAC or physical layer protocols, which contributes to its flexibility for use along with different architectures. However, the performance of the system is only measured through simulation experiments and the correctness of the state diagrams associated with the agents' operation is yet to be verified. Moreover, the overhead for the relatively extensive set of control packets in inter-node communications is not evaluated.

4.1.3 Information-Driven Sensor Querying (IDSQ) and Constrained Anisotropic Diffusion Routing (CADR)

The IDSQ (Chu et al., 2002) and CADR (Zhao et al., 2002) take on an information-driven approach to tracking targets and directing queries respectively.

A leader node, elected during the initialisation phase, waits for an activation prompt from the user/application. Other nodes, referred to as *follower*s, on the other hand, are triggered into their 'sense and report' activity by a request from the leader. An initial *belief*, representing the state of the target, is calculated and the leader begins to keep track of which sensor's measurements get incorporated into the belief state. The leader node is finished processing only if the resultant belief fits some predefined measure of quality; otherwise, it continues with sensor selection. A sample test used within the experiments reported in Chu et al. (2002) for assessing the goodness of a given belief is that the incorporation of measurements is continued until all sensors in a cluster of seven nodes are incorporated.

During the sensor selection phase, the leader chooses a node whose measurements satisfy some quality metric and are not already incorporated into the belief state. For the sake of experiments conducted in IDSQ, four different criteria have been adopted for choosing this next sensor: *nearest neighbour data fusion*, *Mahalanobis distance*, *Maximum Likelihood* and the *best feasible region*. The mathematical details concerning the representations of these criteria can be found in Chu et al. (2002). Once the best next node is selected, it is queried by the leader and the belief state is updated to reflect the newly captured information. The belief quality test is repeated until a desired level is reached. In Liu et al. (2003), a group formation algorithm is proposed in addition which extends the IDSQ method to track multiple targets. In this work, a dynamic group management method is used to initiate and maintain multiple tracks in a distributed manner and each group is responsible for tracking a single target.

Probably, the main drawback to IDSQ is that it relies on a single node (the one closest to the target) to detect the status of the target and does not account for nodes' collaboration to enable simultaneous detection. Hence, the accuracy of tracking in the next steps is highly dependent on the correctness of the initial leader election. The specifics of the leader election algorithm are also not clarified for cases which multiple nodes detect the target.

4.1.4 Distributed Event Localisation and Tracking Algorithm (DELTA)

The DELTA method, discussed in Walchli et al. (2007), is based on dynamic group formation among detecting nodes. It assumes that sensors are aware of their locations and are arranged in a grid structure. DELTA leverages on a passive heartbeat mechanism to relax the assumption that the communication range should be greater than the sensing range. In particular, the leader node is supposed to periodically broadcast HEARTBEAT messages, and the nodes within a two hop neighbourhood are expected to overhear these messages along with their data so as to become aware of the existence of the leader.

Light (illumination) sensors, upon detecting a target, execute a leader election algorithm at the initial phase of the system's operation. A particular node's state may be either of: IDLE, ELECTION_RUNNING, LEADER, and MEMBER and transitions occur as per the exchange of: HEARTBEAT, IREP, PASSIVE and REELECTION messages. The leader node runs a localisation algorithm to determine the position of the target and is required to maintain its leading state as long as possible to minimise the number of reelections and handovers.

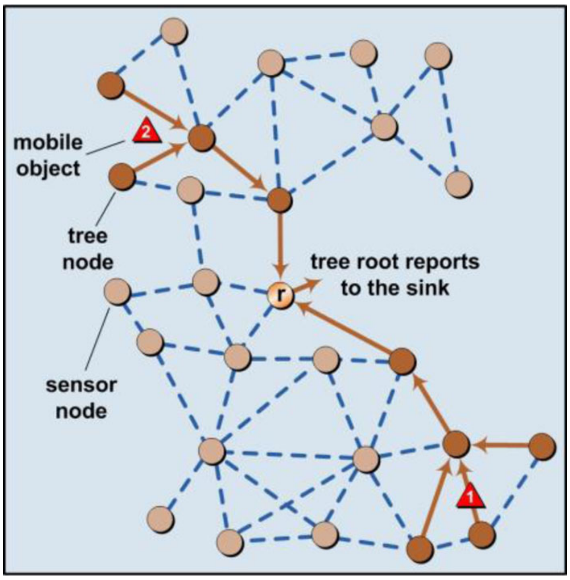
DELTA is primarily intended for single target tracking and it is not explicitly envisaged in Walchli et al. (2007) as to how the associated state diagrams should be modified to account for multi-target scenarios. Moreover, no specific mechanism is envisioned for properly responding to target loss or even to cater for high velocity target tracking. The only reliability-bound considerations in DELTA are that the leader should be the closest node to the target, must have enough battery, and that the election process needs to be carried out within a small time interval. The correctness of DELTA is not verified through formal proofs and its performance is only compared against that of the EnviroTrack system (Abdelzaher et al., 2004) through simulation experiments. What mainly distinguish DELTA from EnviroTrack are its luminosity-based leader election and the specifics of its localisation algorithm.

4.2 Tree-based tracking methods

Tree-based tracking, as discussed earlier in Section 3, features an underlying tree structure which is set up amongst nodes upon target detection. Figure 5 shows a tree-based tracking network where every detecting node hands in its information of the object to its parent, which is responsible for routing the results towards the root. With tree-based substrates comes the advantage of establishing unique loop-free paths from detecting nodes back to the sink, effectively serving as a routing structure as well (e.g., in methods Zhang and Cao, 2004; Tran and Yang, 2006). Probably, the main drawback typical of this category is the large message exchange overhead associated with structure set-up and maintenance which might in turn give rise to scalability issues especially in large-scale or multi-target scenarios. For instance, it is shown in Wang et al. (2010) through simulation experiments that the tree-based tracking protocol presented in Zhang and Cao (2004) has the highest energy consumption compared to its counterparts in Yang and Sikdar (2003) and Wang et al. (2010). Relying on unique paths for source-sink communications might also pose as a downside in case of recurrent link or node failures.

In this subsection, we discuss briefly five tree-based tracking protocols: DCTC, STUN-DAB (with its extended versions: DAT and Z-DAT), OCO, a centralised heuristic method, and Trail.

Figure 5 A tree-based Multi-Target Tracking network: detecting nodes send their information to their parents which subsequently deliver the object's information to the root (see online version for colours)



4.2.1 *Dynamic Convoy Tree-based Collaboration (DCTC)*

In Zhang and Cao (2004), the network is viewed as a grid structure and it is assumed that nodes adopt the Geographic Adaptive Fidelity (GAF) protocol (Xu et al., 2001) as their sleep scheduling mechanism. Each pair of nodes in the neighbouring grids can communicate directly with each other and unless there is a target in the vicinity of a grid, only the grid head is required to be awake and other nodes only need to wake up periodically. Every node is assumed to have a global view of its grid (e.g., other nodes' location information) thanks to the GAF discovery process. DCTC proffers a dynamic programming-based algorithm for constructing an optimal tree (min-cost convoy tree) which assumes that the target's moving trace is known a priori and that each node has global knowledge of the network topology.

In the distributed version of the min-cost convoy tree algorithm, when the target enters the monitoring region, an initial convoy tree is constructed for which a root node has to be determined. A two phase algorithm for determining the root of the tree and a parent for each node is presented which relies on the controlled communication of special *election* and *winner* messages amongst the detecting nodes. Sensors hand in their information of the target to their parents till the results are finally delivered to the root.

Two different member recruitment policy, namely *conservative* and *prediction*-based are proposed in Zhang and Cao (2004) in order for the root to enhance the tree structure by pruning distant and adding nearby nodes. Simulation results in Zhang and Cao (2004) suggest that the prediction-based recruitment scheme outperforms its conservative counterpart, and that it can achieve a relatively high coverage and low energy consumption close to an optimal solution.

In case the target moves away from the current root more than a specified distance, the tree is supposed to be reconstructed and a new node is to be elected as the new root. Tree reconfiguration is also carried out using either a *sequential* or a *localised* approach. In brief, in the sequential approach, as the name suggests, a special *reconf* message is broadcast by the new root to the nodes within its own grid as well as to the heads of its neighbouring grids. A *reconf* message contains node-specific information such as the node's location and the cost of sending data to the root. These grid heads repeat the same process until all nodes in the monitoring region are added to the tree.

As argued in Zhang and Cao (2004), the sequential reconfiguration algorithm has some drawbacks. During each reconfiguration process, large lists containing the information about all nodes in a grid are transmitted between and rebroadcast within the grids. This creates significant amount of traffic, as shown in Wang et al. (2010), especially when the node density is very high or reconfiguration is frequent. The localised reconfiguration, however, removes the cost information from the *reconf* messages and instead makes use of a heuristic to estimate the cost. It also removes the information about the node locations by caching the location information about other nodes for a particular duration. DCTC's more optimised tree reconfiguration schemes are discussed in Zhang and Cao (2004).

In sum, DCTC's operation is heavily dependent on the knowledge of distance to the centre of the event at sensor nodes, which may not always be easy to compute from the sensed information and depends on the accuracy of the mobility prediction algorithms as stated in Xu et al. (2004). Furthermore, it relies on costly message exchanges and broadcasts which adversely affects the system's performance especially when the data rate or target speed is high. Adding and removing distant nodes is also done via the root node, essentially requiring periodic transmissions from leaf nodes towards the root and vice versa.

4.2.2 *Scalable Tracking Using Networked Sensors (STUN) and Drain-And-Balance (DAB) tree*

The STUN architecture (Kung and Vlah, 2003) is founded on a hierarchical structure which allows the system to handle a large number of tracked objects. It is assumed that targets' movements are not uniform across the region and occur in a limited fashion in adherence to the concept of *locality of movements* (Cerpa et al., 2001; Kung and Vlah, 2003). The underlying structure is basically a tree and given the restricted nature of movements, the maintenance of object-related information (e.g., object IDs) usually requires updating nodes near the bottom of the tree. This is done via the concept of *detected set*, in which each intermediate node in the tree stores the set of objects that were detected jointly by its descendants.

The DAB algorithm discussed in Kung and Vlah (2003) is one way to construct the STUN's hierarchical communications substrate. Assuming that the frequencies of object movements are known a priori, DAB assigns weights to edges of the graph of nodes in such a way that if two vertices have more target movement in between, their connecting edge has a higher weight. Nodes with larger weights on their edges are merged together more rapidly – almost akin to the construction of *Huffman* trees (Cormen et al., 1990). The merge process is performed bottom-up, i.e., from the leaves to the root, through a series of steps. The DAB algorithm attempts to strike a balance between communications cost and delay by a set of *threshold factors* that determines the number of steps for merging nodes into the tree. For instance, in the first step with a threshold factor of 6, the vertices on which edges weighing more than 6 are incident are merged together.

Neither the specifics of obtaining the frequency of object movements nor the details of a distributed tree construction are elaborated in Kung and Vlah (2003). How the set of threshold factors, an integral part of the algorithm, is determined is also not clearly described and the paper does not go much beyond giving a numerical example. The DAB tree is essentially a logical overlay which may not reflect the physical structure of the network; hence, a single edge is likely to correspond to multiple communication hops with high costs (Lin et al., 2006).

Extensions to DAB, namely DAT and Z-DAT, are proposed in Lin et al. (2006) with the prime intention of establishing a stronger correlation with the physical topology. In particular, the sensing area is divided into square-like zones and, as opposed to the merging of nodes in DAB, the algorithm recursively combines these zones into a tree. In both DAT and Z-DAT, the query cost of an object tracking tree is formulated in terms of the *query rates* of the sensors (i.e., the number of queries sent towards a specific sensor within a single time unit). Much in the same way as in DAB, Z-DAT relies on a central unit for storing the network topology, which might reduce the scalability and debilitate the distributed capacity of the algorithm, as mentioned in Lin et al. (2009).

4.2.3 Optimised Communication and Organisation (OCO) for target tracking

OCO (Tran and Yang, 2006) is a tree-based target tracking method that provides self-organising and routing capabilities together with the promise of low computation overhead on sensor nodes. OCO's operation is organised into four phases, namely: *position collection*, *processing*, *tracking* and *maintenance*.

Over the course of the initial phase, the base station solicits the position information of all the reachable nodes in the network by contacting its immediate neighbours, gathering their IDs as well as advertising itself as their parents. This process repeats network wide. Following this phase, a tree structure is built across the network in the processing phase within three steps:

- Cleaning up the redundant nodes so as to inactivate those nodes whose sensing coverage region are occupied by one or more other nodes
- Identifying the border nodes for activation
- Finding the shortest path from each node to the base.

During the tracking phase, two types of sensors are exploited: type *A* and *B*. The former has the capability to sense multiple objects and activates its neighbours when a particular object is leaving its coverage area; whereas, type *B* sensors are only able to determine whether there is any object within their coverage area or not. The maintenance phase is intended for the reconfiguration of the network in the face of topological perturbations induced for instance on account of exhausted, damaged or re-positioned nodes.

A major drawback of the OCO method lies in the relatively massive volume of control messages which have to be exchanged during all four phases of the algorithm. Furthermore, all processing functionalities (e.g., deactivating redundant nodes, computing the SPT, etc.) are bound to be done in the base station node (or the root) and only the results would be dispatched towards the specified nodes.

4.2.4 A heuristic tree-based algorithm for tracking

The heuristic tree construction algorithm presented in Lee et al. (2006) is based on the DAB algorithm (Kung and Vlah, 2003) discussed earlier. The network is assumed to be comprised of two types of nodes: *communication* nodes which are only in charge of relaying the information and *sensor* nodes that are capable of detecting the target and sending their information to a communication node. The DAB's weight assignment initiative is generalised in Lee et al. (2006) by factoring in, a 'two-way' object moving frequency between each pair of sensor nodes. Hence, it is able to account for cases where the frequency with which a particular object moves from node x to node y differs from that of the reverse direction. Furthermore, in this extended model, transmission costs are associated to the links.

In Lee et al. (2006), the problem of building a minimum cost object tracking tree is formulated in terms of a "0–1 integer programming" model for which a Lagrangean Relaxation-based (LR-based) heuristic algorithm is proposed.

The optimisation problem centres around three different decision variables: *paths*, *tree links* and *tracking links*. Paths are represented by tuples in the form of (s, *sink*), indicating that a path from sensor node s to the *sink* node exists. It is further assumed that the set of all candidate paths are given as an input to the algorithm. Tree links are the connecting edges on the object tracking tree. Tracking links, however, are the links when object moves from sensor x to sensor y, and then sensor y delivers tracking information upward to the first common ancestor (i.e., the communication node which has nodes x and y in its sub-tree) via the tracking links.

Despite the fact that the protocol is intended to minimise the total communication cost of the tree, the centralised computation of both the LR-based primal heuristic and the object tracking tree algorithms adversely affect its scalability. Despite that centralised optimal solutions can be viewed as a reference baseline for comparison of tracking methods, it is more desirable to come up with near optimal distributed tracking protocols instead.

4.2.5 Trail

Trail (Kulathumani et al., 2007, 2009) builds and maintains a tree-like tracking data structure by propagating the mobile object's information locally and by satisfying the distance sensitive requirements (much the same as DQT (Demirbas and Lu, 2007)). In particular, the data structure associated with an object P, viz., $trail_P$ maintains for P information such as two-dimensional position coordinates p, and its distance from a designated centre C, denoted by d_{pC}. In order to keep from triggering updates every time P makes a move, $trail_p$ is not required to be the exact straight line from C to P; i.e., Trail effectively allows for stretch factors slightly bigger than one, essentially resulting in the recruitment of a path of nodes (line segments in Trail's language) for the tracking mission. A *Find* algorithm is additionally proposed to reach the location of object P initiating from another object Q given that $trail_p$ exists.

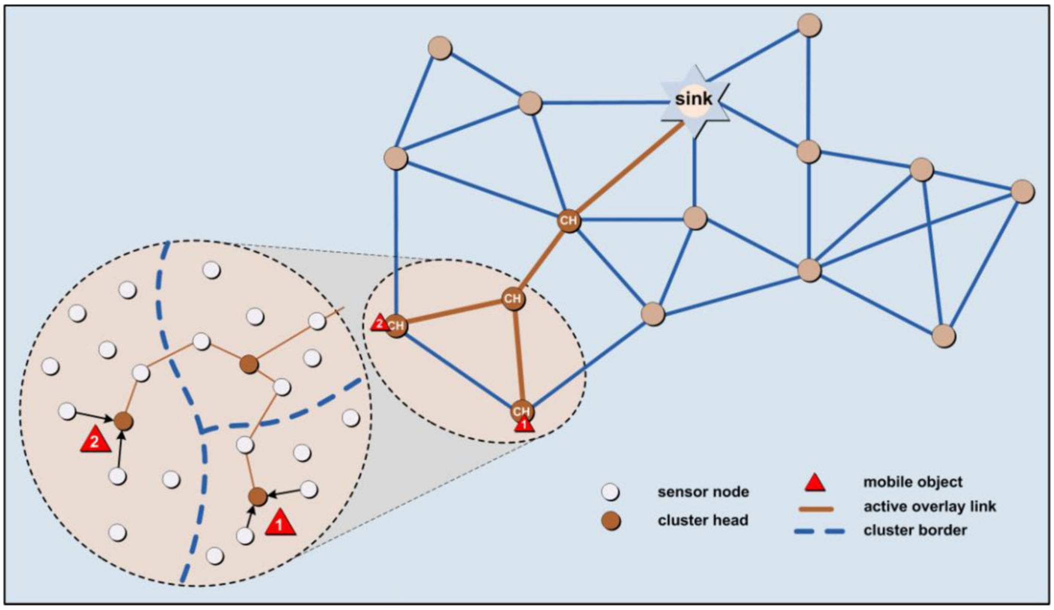
Trail has been implemented in a WSN grid in the form of a discrete plane, though any random deployment of nodes can also be used. Every node is assumed to be aware of its location and maintains a list of its immediate neighbours. Multi-hop communications in Trail draw on an underlying geographic routing protocol such as GPSR. Trail's measures for fault tolerance account for the *update*, maintenance of an existing trail and for its *Find* algorithm, which are claimed to be able to gracefully handle any performance degradation resultant from node failures. Trail's operation can also be refined in terms of the mean time to structure update and tuning up the specifics of the reshaping process.

In Kulathumani et al. (2007), Trail is only contrasted analytically with few analogous tracking protocols, and the simulation experiments are not intended for exhaustive comparison. Trail's operation can be enhanced with predictive models and with support for MTT scenarios.

4.3 Cluster-based tracking methods

In cluster-based tracking protocols, the underlying structure is either proactively set up amongst nodes (Xu et al., 2004; Tsai et al., 2007; Chen et al., 2004; Yang and Sikdar, 2003; Goel and Imielinski, 2001; Xu et al., 2004; Olule et al., 2007; Yu et al., 2004) or forms online upon target detection (Ji et al., 2004; Chang et al., 2008; Wang et al., 2010). Figure 6 presents a hierarchical cluster-based object tracking network, in which nodes in detecting clusters send the targets' information to their associated cluster heads (blue nodes). Cluster heads, in turn, contact via other clusters for relaying data towards the sink or they may be able to communicate directly with the sink node. Furthermore, it is assumed that the cluster structure shown in this figure has been set-up according to some criteria, which we do not go through its details.

Figure 6 A cluster-based Multi-Target Tracking network: detecting nodes send the targets' information to their associated cluster heads which, in turn, contact other clusters for relaying data towards the sink (see online version for colours)



Despite their relatively high maintenance overhead, cluster-based schemes demonstrate acceptable scalability against the number of nodes and/or targets within the monitoring region (Chen et al., 2004; Yang and Sikdar, 2003). Most of the tracking protocols, however, utilise the traditional line of clustering algorithms (e.g., LEACH

(Heinzelman et al., 2002)) or do not rely on a specific clustering algorithm at any level, which may lead to poor performance; for instance, using LEACH (Heinzelman et al., 2002) in Wang et al. (2010), which mainly guarantees the uniform distribution of cluster heads, or using Voronoi diagrams in Chen et al. (2004) and the planar graph in Tsai et al. (2007), which are based on geographic locations of the nodes, are not optimum choices for utilisation in the context of OTSNs. More specifically, over the course of an object tracking mission, the cluster head or the leader node must be the one nearest to the target so that it would be able to provide better estimations of the location of the target and thus cut down on energy consumption (Park, 2006). Only a few number of tracking protocols, such as Ji et al. (2004), Chang et al. (2008), Park (2006) and Wang et al. (2010), feature dynamic clustering based on properties of objects, which are discussed in this section.

Here, we have presented a synopsis of eight cluster-based tracking protocols: DOT, DCS (together with its extended version, CODA), a dynamic clustering method for acoustic target tracking, RARE-Area/RARE-Node, DPT, PES (together with DPR and PREMON), an adaptive target tracking approach, and finally HCTT.

4.3.1 Dynamic Object Tracking (DOT)

DOT (Tsai et al., 2007) features a cluster-like tracking structure in the form a planar graph which is constructed amongst nodes before a target enters the area. This graph divides the network into a number of *faces* in which nodes *u* and *v* are deemed face neighbours only if they are connected via a *Gabriel* edge. DOT's basic operation consists of six phases: *face discovery*, *target discovery*, *target detection*, *target tracking*, *face-track shortening* and *loop face-track removing*.

Once the planar graph is established in the initial phase (face discovery) and all nodes get to know their face neighbours, a user may seek a particular target by flooding its request throughout the network. In the target discovery phase, a node receiving a request replies only if it is the nearest node to the target and if the target is likely to remain in its face. As for the target detection phase, the nearest node to the target is considered as a *beacon* or *ingress* node which awakes its face neighbours by sending *wakeup* messages. Every active node that detects the target must, in turn, verify whether it is the nearest node or not. Within the target tracking phase, when a target moving across a given face reaches another node, the ingress might change and the new nearest node should instead send wakeup messages. In case the target exits a face, the current beacon node records the next beacon and informs the user of this transition.

The face-track shortening phase is intended for tracking high velocity targets. The intermediate beacon nodes report the last known location of the target along a shorter path through which the user is able to chase the target with a smaller loss probability. Moreover, when the velocity of the target is high, it might form a loop on its return to a previous node. The sixth phase is, thus, aimed at resolving possible loops within the face-track.

DOT's operation entails a remarkably high number of control packet exchanges, especially during the initial phase of the algorithm; i.e., the computation of the planar graph. Neither a proof of correctness nor an analytical evaluation is presented in Tsai et al. (2007), and the specifics of query propagation are not clearly discussed. Reliability of the algorithm in terms of track maintenance depends much on the frequency of wakeup messages. Additionally, in case the covered areas associated with the graph faces are not large enough, the DOT protocol is susceptible to track loss. The only remedy for this situation, as reported in Tsai et al. (2007) is to rediscover the target by flooding a request to the whole network.

4.3.2 Dynamic Cluster Structure (DCS) and Continuous Object Detection and tracking Algorithm (CODA)

DCS (Ji et al., 2004) and its augmentation, CODA (Chang et al., 2008), deal with continuous object detection. In DCS, the nodes detecting the presence of the object declare themselves as *boundary sensors* only if the object goes undetected by the other nodes in their immediate neighbourhood. Using a location-based approach, a clustered structure is then constructed for grouping these boundary sensors into a small number of dynamic clusters, and the closest node to the centre of each cluster area heads the cluster. The number of boundary sensors in the cluster can be pre-determined. Following the clustering procedure, all boundary sensors of a continuous object are assigned to several clusters according to their locations. Cluster heads send the integrated boundary information to the sink which is supposed to estimate the global boundary of the continuous object of interest.

Whlie it is argued in Chang et al. (2008) that adopting a clustering approach yields a significant reduction in the overall communications overhead compared to schemes in which the sink contacts directly with the individual boundary sensors, DCS still requires a high volume of communications and message exchanges amongst nodes. For instance, the boundary sensors are identified by requiring each detecting sensor to communicate with all its one-hop neighbours to determine whether or not they have detected the same object. Furthermore, clustering the boundary sensors entails repeated broadcasting of messages from cluster heads to their neighbours until a specific hop count is reached. Besides, the number of required clusters and thus, the number of cluster heads increases as the continuous object expands, frequently triggering dynamic re-clustering operations.

CODA (Chang et al., 2008) presents a hybrid semi-dynamic clustering scheme aimed to reduce the bulk of DCS-induced message exchanges and thus to cut down on energy consumption. During the initial stage of the network operation, a static backbone is constructed featuring a designated number of static clusters in which any sensor detecting the object reports the associated information to its cluster head. The cluster heads, in turn, determine the boundary sensors outlining the continuous object within their associated cluster using a local boundary estimation function, and report the results to the sinks. CODA's performance has been evaluated in Chang et al. (2008), with the promise of a significant decrease in the number of control messages, albeit for a fewer number of clusters in comparison with that of DCS. It might however be the case that CODA and DCS perform consistently when tested at identical settings.

4.3.3 Dynamic clustering for acoustic target tracking

In Chen et al. (2004), network nodes are assumed to be either sparsely placed static backbone nodes, taking on the role of cluster heads, or densely deployed low capability sensor nodes used for detecting the target. The system's operation is organised into four main phases: *initial distance calibration and tabulation*, *CH volunteering*, *sensor replying*, and *reporting the tracking results*.

In the calibration phase, the geographical information necessary for the construction of Voronoi diagrams is collected, and both the cluster heads and the sensors establish response tables to facilitate their decision on adjusting back-off timer values. These timers are used when a cluster head intends to volunteer itself as a leader or when a sensor intends to respond to the cluster head.

During the second phase, a cluster head, upon detecting an acoustic signal whose strength exceeds a determined threshold and whose pattern matches the system's tracking interest, volunteers to become active. A two-phase broadcast mechanism is then utilised for setting two back-off timers in accord with the content of the response tables. After the expiration of the first timer, an *energy* packet (that carries only the signal strength information), and after expiration of the second timer, a *signature* packet (that contains the detailed signature information) are broadcast by the volunteer cluster head. Should the CH overhear another energy or signature packet, it cancels its volunteering process.

Following the volunteering process, one cluster head is active and it estimates the target's location after receiving enough replies from sensors in the third phase. Cluster heads report their tracking results to users in a proactive fashion, thereby reducing their experienced latency. No specific routing protocol has been envisaged explicitly in Chen et al. (2004) to back up the message transmission in this phase.

Through both probabilistic analysis and simulation, it is shown in Chen et al. (2004) that with the use of Voronoi diagram, the CH closest to the target stands the greatest chance of getting elected as the leader and that the proposed dynamic clustering algorithm yields relatively accurate estimates of target locations. As for the final phase, the overhead associated with the transmission of reports can be prohibitively large for a greater number of users in that a larger number of messages might need to be routed on distinct paths to reach the base station. Similar to DOT, this method has an initial phase for constructing the Voronoi diagram, calibration and tabulation of the tables which would be carried out offline.

4.3.4 Reduced Area REporting (RARE-Area) and Reduction of Active node REdundancy (RARE-Node)

In Olule et al. (2007) two algorithms, RARE-Area and RARE-Node are presented for increasing the quality and decreasing the amount of data sent from sensing nodes to their cluster heads. The former algorithm ensures that only nodes promising a given quality of data participate in tracking. The latter, on the other hand, requires that any node with redundant information be laid off from the tracking mission. It is assumed that each node knows both its own location and that of its neighbours through the well-known Trilateration localisation algorithm. It is also assumed that a static clustering is performed at the first stage of the network operation.

In RARE-Area, every node that detects a target calculates a weight value (W) as the sum of three parameters: target distance from sensor (D), direction of motion (D_r) and target velocity (V). A node sends a *beacon* message and waits for at least two other beacons from its neighbours only if its computed W exceeds a given threshold, W_U, set at the beginning of the algorithm's operation. Upon receiving enough beacons, the node can estimate the target's location and subsequently forward it to its cluster head.

When RARE-Node is also utilised, a node willing to send its information to the cluster head performs a redundancy check as well by verifying whether there are any other node within its sensing range with redundant information or not, for instance through detecting overlapping sensing areas with neighbouring nodes. If the target's position is estimated to be within a node's half space of the overlapping region with a neighbour, only this node sends its information to the cluster head, sparing the neighbour from all redundant chores.

RARE-Area and RARE-Node algorithms are more advantageous in networks or clusters with large numbers of nodes; however, the experiments reported in Olule et al. (2007) assume a network of one cluster head and 49 sensing nodes, which evaluates the algorithms only at node level and not within a hierarchical structure. Hence, the usefulness of the static clustering at the initial phase of these algorithms is yet to be investigated.

4.3.5 Distributed Predictive Tracking (DPT)

In DPT (Yang and Sikdar, 2003), it is assumed that a cluster structure is already established amongst nodes with every cluster head knowing the ID, location and the remaining energy of nodes under its management. The main idea behind DPT is that upon target detection, a sequence of tasks in order of "*sense-predict-communicate-sense*" be executed in a distributed fashion by the nodes along the target's track. The algorithm consists of four major procedures: *target descriptor formulation*, *sensor selection*, *failure recovery*, and *energy considerations*.

Each target is identified by a *Target Descriptor* (TD) representing its identity, its present location, its next predicted location, and a time stamp. A cluster head utilises a target descriptor formulation algorithm to obtain the TD for each target it is tracking. The sequence of downstream cluster heads lying along the target's track (CH_1, CH_2, …, CH_N) is assumed to be known a priori by the cluster heads. Besides, each TD is sent all the way back to the sink for further processing as well as to the downstream CH, i.e., a CH_{i+1}. The third field in TD, the target's next predicted location, is essentially a linear predictor which only uses the previous two locations to linearly predict a possible future location.

For the purpose of the sensor selection algorithm, i.e., to locate the target's position, at least three nodes must detect the target. When the target enters a downstream cluster, the CH within this cluster (CH_{i+1}) is supposed to activate three nearest nodes to the target via a wake-up message. These three nodes, in turn, would send their information of the target to CH_{i+1} in the form of a location message. CH_{i+1}, then, formulates TD_{i+1} and estimates the next location of the target based on the received information, current direction and the target's velocity. The target's next location is sent for the next nearest CH and the process is continued. In case a CH is unable to find three sensors not farther than its low beam (r) to the target of interest, it would expand its search radius to look for eligible sensors within its higher sensing beam (R). Should CH_{i+1} fail in its attempt to find enough sensors even using its high sensing beams, it calls for assistance from its neighbouring cluster heads.

The failure recovery process is intended to account for two situations: link/node failures and prediction errors. The process is organised into several levels, where in the Nth level, a group of sensors $(2N - 3)r$ m away from the ith predicted location of the target would be activated for detection. Simulation results reported in Yang and Sikdar (2003) only suggest that the first level outperforms the other recovery levels in terms of energy savings, but no further explanation regarding the specifics of the recovery formula, nor a proof of correctness has been given as to the algorithm's capability in detecting missing targets.

The adoption of strategies such as sending activation messages together with the conservative sensor selection policy using normal and high beams for sensing forms the mainstay of DPT's energy considerations.

Although DPT is among the very first methods primarily designed to exploit a prediction strategy, its simulation results on miss probability and energy consumption are not promising when it comes to tracking high velocity targets. Furthermore, the simplifying assumptions like that each cluster head knows the location of all its peers and that each TD is sent back to the sink come at the expense of high overhead.

4.3.6 Prediction-based Energy Savings (PES), Dual Prediction-based Reporting (DPR) and PREdiction-based MONitoring (PREMON)

Achieving energy savings by enabling the sensor nodes to intelligently predict the future movements of the mobile object can be deemed as the main drive in Xu et al. (2004), Goel and Imielinski (2001) and Xu et al. (2004). In all three schemes, it is assumed that a cluster structure exists among nodes, although no specific clustering mechanism is prescribed for the algorithm's operation.

PES (Xu et al., 2004) is comprised of three main components: a *prediction model*, a *wake-up strategy* and a *recovery mechanism*. The prediction model anticipates the future movement of an object to activate only the sensor nodes expected to discover the object. The wake-up strategy resorts to some heuristics to balance energy savings and application requirements, effectively arranging which nodes and when they should be activated. The recovery mechanism is initiated only in the event that the network loses the track of an object.

Based on the prediction model used, the *current node*, i.e., a node with a moving object within its territory, predicts the possible location(s) of the target and determines a group of sensor node(s), viz., *target nodes*, to help track the target after a certain period of sleeping. Three heuristics have been introduced in the prediction model of PES based on the speed and direction of the target: INSTANT, AVERAGE and EXP_AVG. For instance, using the INSTANT heuristic, the current node assumes that the target maintains its current speed and direction for the next $(T - X)$ seconds (T and $X < T$ are the reporting and sampling durations resp.). In addition, target nodes are chosen based on either of the DESTINATION, ROUTE and ALL_NBR heuristics. With the DESTINATION heuristic, for instance, the current node only informs the predicted destination node, i.e., the sensor node where the object eventually arrives.

The current node inactivates itself once it sends the wake-up call to the target node(s). As argued in Xu et al. (2004), no matter what heuristic for prediction/activation is used, the miss rate of the target is non-zero, which warrants a recovery mechanism for the system's operation. The recovery process in PES initially resorts to the ALL_NBR heuristic to activate all neighbouring nodes of the current node. Unless the target is found, the next procedure in the recovery mechanism is to trigger a *flooding recovery* which wakes up all nodes in the network.

In DPR, both the sensor nodes and the base station make predictions about the future movements of the mobile objects (Xu et al., 2004). The first time prediction is made on the basis of an initial given value, while the subsequent periodic predictions draw on a PES-like model together with the objects' movement history. Through monitoring the state of the mobile objects, the sensor nodes are able to verify if the predictions are consistent with the actual target movements, and to decide whether to send updates for correcting the inaccurate predictions at the base station or not.

PREMON's operation (Goel and Imielinski, 2001) is almost analogous to that of DPR's with the only difference that the sensor nodes do not make predictions and only monitor the state of the object. It is assumed that sensors in close proximity are likely to have correlated readings and that the base station is able to predict the sensor readings given certain historical and background knowledge. These predictions are represented concisely as a 'prediction-model' and sent to the appropriate sensors. On receiving a prediction-model, sensors modify their behaviour; i.e., rather than sending an update whenever their reading changes, an update is sent only in case of inconsistencies with the prediction model.

The specifics of the cluster structure and how they affect the systems' operation are not treated pivotal in both PES and DPR; in effect, these algorithms mostly function at the individual sensor node level, much the same as is the case with the RARE system (Olule et al., 2007). Moreover, it is argued in Xu et al. (2004) that the combination of PES and DPR as a single model yields promising results w.r.t. energy savings. PREMON, on the other hand, operates mostly in a centralised fashion where the base station predicts the future sensor readings, given the previous knowledge along with the current information, and it transmits every prediction to the corresponding sensor. According to the results derived from simulation experiments in Xu et al. (2004), DPR outperforms PREMON in terms of energy consumption.

4.3.7 Adaptive target tracking

A quality aware information collection protocol is proposed in Yu et al. (2004) for WSN tracking. Similar to the case in PES, DPR and PREMON, the adaptive tracking method mostly functions at the individual node level, requiring that sensor read-outs be sent separately to the sink. However, a cluster-based operation can also be envisaged for the system as argued in Yu et al. (2004). Central to the algorithm is a sensor/server model featuring a state diagram of three operational states for the nodes: *active*, *quasi-active* and *monitor*. The server calculates the location of the target using the well-known Triangulation algorithm.

The sensor nodes sense the environment and communicate with the server within a period of t_{sense} and t_{send}, respectively. The decision on sending an update message to the server is dependent upon both a node's current state and the likely occurrence of any external event. In Yu et al. (2004), two quality parameters have been defined with the purpose of boosting the accuracy of sensor read-outs: *track quality* ($\varepsilon_{\text{track}}$), representing the maximum distance between the real track of the target and the approximate track generated by the tracking algorithm at the server side, and *sensor measurement quality* ($|\Delta I|$), which is the maximum divergence between a sensor's measurements series at time t and its approximation. The values of both parameters are calculated at the server side and the resultant $\varepsilon_{\text{track}}$ is specified as the desired application tolerance. The server alters the state of an active node only if it realises that the readings received from that sensor render the estimated location of the object more erroneous than the desired $\varepsilon_{\text{track}}$. The sensors are notified of the desired quality level through the value of $|\Delta I|$.

The algorithm is essentially characterised by a centralised logic, and the list of active and quasi-active nodes at the server are maintained in adherence to the readings received from sensors and according to the value of $\varepsilon_{\text{track}}$. Another drawback of this tracking protocol is that in order to set the values of $\varepsilon_{\text{track}}$ and $|\Delta I|$, the server needs to be armed with precise information on the object(s) movement, an assumption which is not realistic in most tracking applications.

4.3.8 Hybrid Cluster-based Target Tracking (HCTT)

HCTT (Wang et al., 2010) features a hybrid clustering strategy which operates by forming dynamic clusters on top of an existing static structure. HCTT assumes that a suitable clustering substrate already exists in the network and that each node is aware of its own location as well as those of its neighbours. Each node v_i needs to be equipped with an acoustic sensor of range r_s whose monitoring region is denoted by the disk $R(v_i, r_s)$. A dynamic cluster forms and temporarily takes charge of the tracking operation whenever the targets' movements reach the boundary of the current static cluster.

Once the static cluster is established, HCTT's dynamic operation can be characterised in terms of three main phases: *boundary node formation, dynamic clustering* and *inter-cluster handoff.*

Over the course of the initial phase, each node consults with its neighbour list looking up for a node belonging to another cluster; in case such a neighbour exists, the node considers itself of type *boundary;* otherwise, it is an *internal* node. Following this step, each cluster is divided into three regions: *safety*, *alert* and boundary. The safety region of a given cluster is to be monitored by at least one internal node, but not by any boundary node. The monitoring of the *boundary region*, on the other hand, is handled by the boundary nodes within the cluster together with those of the adjacent clusters at the same time; accordingly, the *alert region* is monitored by any boundary node of a cluster not belonging to the boundary region of that cluster.

The dynamic clustering phase is triggered whenever a target moves from one static cluster to another, or alternatively when the target moves along the boundaries of one static cluster. When the target is in the alert region, a dynamic cluster is constructed in HCTT within three steps: leader selection, dynamic cluster construction and boundary node formation. During these steps, the nearest node to the target is selected as the leader which subsequently broadcasts a *recruit* message to its neighbours. Those neighbouring nodes replying with a *confirm* message participate in the new dynamic cluster. Once the new cluster is constructed and upon the target's entrance to the next region, the inter-cluster handoff process is triggered. The specifics of the handoff process varies depending on whether the transition occurs from a static to a dynamic cluster, from a dynamic to a static cluster, or alternatively from one dynamic cluster to another.

The synergistic exploitation of dynamic and static clusters effectively renders HCTT immune to target loss (i.e., there always exists a cluster which keeps track of the target), however, it comes at the expense of heavy message exchanges and a high level of energy consumption also evidenced by the outcome of the experimental evaluations reported in Wang et al. (2010). For instance, a single static to dynamic inter-cluster handoff, alone, triggers the exchange of six types of control messages, i.e., *request*, *reply*, *work*, *sleep*, *resign*, and *dismiss*. The dynamic cluster construction in the alert region also requires a costly setup. A promising direction to reduce the overhead of control messages in HCTT is to somehow place the dynamic clusters more intelligently along the direction of target movements.

5 Summary and discussion

In this section, we summarise our review of the prior art and point out to which extent these systems are well-suited for a mobile object tracking scenario. Our discussion commences with a rough comparison of the tracking schemes when only viewed from the network structure standpoint, and continues with the exploration of their similarities and differences in terms of: strategies for deactivating non-necessary nodes, target recovery mechanisms, MTT support, routing/querying issues, as well as the specialised techniques for data aggregation and localisation services. Table 3 recapitulates our summary.

Structure maintenance over the course of the tracking operation is an integral facet of both cluster and tree-based methods, and also gives rise to a trade-off between precision and robustness on one hand and control overhead on the other. In general, it can be argued that compared to cluster and leader-based methods, tree-based algorithms incur the largest overhead in terms of both building/maintaining a communication substrate as well as source-to-sink transmissions across multi-hop paths (Wang et al., 2010). Furthermore, structured systems are subject to significant increase in energy consumption as the network scales up, as the monitoring area expands or as the number of targets increases. In DCTC (Zhang and Cao, 2004), for instance, control operations draw on heavy message exchanges and broadcasts which might degrade the system's performance, especially when the data rate and/or the target speed is high. As discussed earlier, the overhead associated with the tree-based communications in OCO (Tran and Yang, 2006) and in HCTT (Wang et al., 2010) can also be overwhelming.

Table 3 Summary of object tracking methods

	Metric			*Protocol overhead*				
Network-centric tracking techniques	*Method*	*Deactivating non-necessary nodes*	*Recovery mechanism*	*Pre-processing cost*	*Minimum number of control packets/ other control overhead*	*MTT support*	*Routing (R)/ Querying (Q) or Aggregation (A) protocol*	*Localisation algorithm or initiative*
Leader-based	LESOP	✓	×	×	3	✓ (with TDMA)	×	Linear combination of sensors' location
	Mobile agent-based (Tseng et al., 2004)	✓	✓	✓	7	✓	✓ (A) TB & TD methods	Trilateration
	IDSQ	✓	×	×	4	×	✓ (R) CADR	Nearest sensor location to the target
	DELTA	×	×	×	5	×	×	CH computes target's location
Tree-based	DCTC	✓	×	×	6	×	✓ (R and Q)	Nearest sensor location to the target
	STUN-DAB & DAT & Z-DAT	×	×	✓	1+ Centralised tree construction	✓	✓ (R and Q)	Nearest sensor location to the target by root
	OCO	✓	✓	✓	5	✓	✓ (R and Q)	BS, as the nearest node, computes target's location
	Heuristic method (Lee et al., 2006)	×	×	✓	Centralised tree construction	✓	✓ (R and Q)	Nearest sensor location to the target by root
	Trail	×	×	×	5	×	✓ (find-centric Trail)	Nearest sensor location to the target

Table 3 Summary of object tracking methods (continued)

				Protocol overhead				
Network-centric tracking techniques	*Metric* *Method*	*Deactivating non-necessary nodes*	*Recovery mechanism*	*Pre-processing cost*	*Minimum number of control packets/ other control overhead*	*MTT support*	*Routing (R)/ Querying (Q) or Aggregation (A) protocol*	*Localisation algorithm or initiative*
Cluster-based	DCS & CODA	✓	×	DCS × CODA ✓	2 + cluster construction	Continuous object	×	Sink computes object's boundary
	Acoustic tracking (Chen et al., 2004)	✓	×	✓	5	✓	×	Voronoi diagram-based and nonlinear optimisation-based
	DOT	✓	×	✓	8 + planar graph construction	✓	×	Trilateration
	RARE (Node & Area)	✓	×	✓	2 + static cluster construction	×	×	Trilateration
	DPT	✓	✓	✓	8 + cluster construction	✓	×	Linear prediction
	PES & DPR	✓	✓	✓	6 + cluster construction	✓ ✓	×	BS computes target's location
	PREMON	✓	×	✓	3 + cluster construction	×	×	BS computes target's location
	Adaptive tracking (Yu et al., 2004)	✓	×	× (✓ if clustering is applied)	3 + cluster construction (if applied)	×	×	Triangulation (server computes)
	HCTT	✓	×	✓ (the static clustering)	16 + static cluster construction	×	×	No specific methods (all members participate the estimation)

As can be noted from Table 3, most tracking mechanisms leverage on special structures and/or make use of some prediction strategy in order to deactivate non-necessary nodes to reduce energy consumption. For instance, the RARE algorithm (Olule et al., 2007) relies on a static cluster structure built before the launch of tracking-bound operations; whereas, the structure-less LESOP (Song and Hatzinakos, 2007) draws on a prediction-based scheme to determine the next leader while leaving irrelevant nodes in sleep mode. DPT (Yang and Sikdar, 2003), PES (Xu et al., 2004) and DPR (Xu et al., 2004), as a hybrid system of a prediction-based scheme together with a clustered structure, would incur relatively smaller energy overhead. DOT (Tsai et al., 2007) and the mobile agent-based method discussed in Tseng et al. (2004), on the other hand, leverage on special structures to reserve unnecessary nodes in sleep mode, and thus, their performance would be comparable to that of the cluster-based methods when it is viewed from the energy usage perspective.

The recovery mechanism, used in response to target loss, can be deemed as the other distinguishing factors when it comes to compare different tracking methods. Although the prediction-based schemes (e.g., DPT (Yang and Sikdar, 2003), PES (Xu et al., 2004) and DPR (Xu et al., 2004)) do not enjoy total immunity to target loss, but its probability is minimised compared to when no prediction strategy is applied. In these schemes, a recovery mechanism is also assimilated which is supposed to take over every time a target gets lost. An explicit recovery mechanism is also devised in Tseng et al. (2004), Tran and Yang (2006). In DOT (Tsai et al., 2007), increasing the nodes' wakeup frequency reduces the target loss probability, and in DCTC (Zhang and Cao, 2004) as well as DAB (Kung and Vlah, 2003), this reduction is achieved through increasing the node density in the monitoring region and by constructing trees/clusters over the densely covered area. The issue of target loss is not explicitly tackled with in DOT (Tsai et al., 2007) and Trail (Kulathumani et al., 2007), but instead the applicability of a solution for the parallel problem of *hole*/*obstacle* has been investigated through the use of a planar graph in Tsai et al. (2007). HCTT (Wang et al., 2010) experiences nearly no target loss since the target is monitored either by a static or a dynamic cluster; hence, no recovery mechanism is provided for the system.

In terms of control overhead, most tree and cluster-based methods, such as DOT (Tsai et al., 2007), DPT (Yang and Sikdar, 2003), OCO (Tran and Yang, 2006) and HCTT (Wang et al., 2010) entail an energy consuming pre-processing stage. While the minimum number of explicit control messages for DCTC (Zhang and Cao, 2004), DOT (Tsai et al., 2007), the mobile agent-based method in Tseng et al. (2004) and DELTA (Walchli et al., 2007) is given in Table 3, for some cluster-based schemes (e.g., PES (Xu et al., 2004), DPT (Yang and Sikdar, 2003), PREMON (Goel and Imielinski, 2001), DPR (Xu et al., 2004), RARE (Olule et al., 2007) and HCTT (Wang et al., 2010)) no specific clustering algorithm is stipulated, and thus, the exact number of control packets cannot be determined. In another line of work, STUN and DAB (Kung and Vlah, 2003), their extensions, DAT and Z-DAT (Lin et al., 2006), as well as the heuristic method discussed in Lee et al. (2006) are listed as 'centralised' tracking systems in Table 3.

As with the support for MTT, some of the schemes listed in Table 3 cater for MTT as well. In some methods, however, the MTT capability is not explicitly supported, but rather provisions have been made to open up for future extensions; for instance in Song and Hatzinakos (2007), orthogonal radio channels are needed in the algorithm to support MTT; a feature that can be provided using TDMA or FDMA channel allocation schemes. In Chen et al. (2004), drawing on signature information increases the system's robustness when dealing with multiple targets. In Liu et al. (2003), a group formation algorithm is proposed to enhance IDSQ (Chu et al., 2002) with the MTT functionality. The STUN structure and its DAB algorithm (Kung and Vlah, 2003) are mainly proposed for environments with a large number of targets (e.g., humans in urban areas). In methods, such as DCTC (Zhang and Cao, 2004), RARE (Olule et al., 2007), PREMON (Goel and Imielinski, 2001), adaptive tracking (Yu et al., 2004), DELTA (Walchli et al., 2007), HCTT (Wang et al., 2010) and Trail (Kulathumani et al., 2007), no explicit support for tracking more than one object has been envisaged.

The routing algorithm used for sending the tracking reports to the sink is another important consideration in assessing target tracking solutions. In the mobile agent-based method discussed in Tseng et al. (2004), two different tactics, namely TB and DB, are presented. In the former, the master node decides whether to send its report to the gateway (sink), and in the latter, it waits and carries the results until a special condition is met. CADR (Liu et al., 2005) and IDSQ (Chu et al., 2002) feature an information-driven routing strategy for directing queries to the source of information and vice versa. The Trail's find-centric approach can also be used as a querying mechanism (Kulathumani et al., 2007). The STUN structure (Kung and Vlah, 2003), DAT/Z-DAT (Lin et al., 2006), OCO (Tran and Yang, 2006) and the heuristic method in Lee et al. (2006) all make use of a tree structure for the transmission of both queries and results. In most tracking methods, however, the specifics of the routing decisions are avoided as an issue, and instead tracking reports are simply assumed to be delivered to the sink(s) via a single-hop connection or else through using some multi-hop routing algorithm.

Localisation algorithms as well as the nodes' location information are also indispensable to the tracking systems. It can be viewed as a mandatory assumption that nodes know their location and that this knowledge be exchanged amongst immediate neighbours. The location information can either be set in nodes a priori, or it can be acquired using a localisation algorithm whose overhead essentially adds up to the preprocessing stage. In part of the existing schemes, a central node (e.g., 'root' in OCO (Tran and Yang, 2006) and Lee et al. (2006), 'sink' in DCS (Ji et al., 2004) and CODA (Chang et al., 2008), 'BS' in PES (Xu et al., 2004), DPR (Xu et al., 2004) and PREMON (Goel and Imielinski, 2001)) which is aware of the locations of all nodes, computes the continuous target's boundary or the discrete target's whereabouts.

As a final remark, it is worth mentioning that we had no intention to come up with an exhaustive comparative study in this section; in effect, Table 3 could still be extended to account for contrasting object tracking systems w.r.t. their prediction strategies, architecture or style of operation (e.g., centralised vs. distributed), the specifics of their underlying structure (e.g., static vs. dynamic clustering), and the sensitivity of the algorithms in the face of varying object directions and speeds.

6 Open issues and research directions

Object tracking has recently emerged as a significant part of the WSN's mainstream of research; however, many aspects of the problem still give rise to a number of interesting open issues, some of which are discussed in this section.

As new object tracking methods are contributed to the existing literature, classification with reference to a richer set of criteria becomes necessary. On the basis of the functioning layer, for instance, some methods may operate at the network level while others are particularly intended for higher layers, or even as is the case with LESOP (Song and Hatzinakos, 2007), the proposed scheme might be a multi-layer architecture with each layer addressing a specific aspect of the tracking problem. We characterised a complete object tracking solution as a synergistic system of application-layer CSP-based algorithms and network/transport-layer communications-oriented protocols; however, the investigation and classification of tracking systems through the prism of CSP-based concepts, such as data fusion strategies, estimation algorithms, identity management (Oh et al., 2005; Shin et al., 2003), etc. call for a dedicated research study.

When considering the effects of other layers and services, many links to the topic of 'multicast protocols' in sensor networks have also to be taken into account. In this style of communication, also referred to as *Mobicast*, data transmission takes place between groups of nodes. The works discussed in Chen and Ann (2005) and Huang et al. (2004), and more recently in Wang et al. (2008) specifically deal with this issue. In their recent survey in Bhatti and Xu (2009), Bhatti and Xu have considered the "Mobicast message based" tracking algorithms as a separate group in their classification whose main objective is to predict target moving directions and to wake up appropriate nodes before target arrival. Within our perspective, these protocols can be subsumed under either the cluster or leader-based protocols with prediction capability. However, a more in-depth assessment is required to explore the efficacy of these protocols for object tracking scenarios.

As discussed in the previous section, a majority of the existing tracking schemes are subject to target loss and feature no specific recovery mechanism in response. Additionally, in tracking methods equipped with a recovery procedure, such as DOT (Tsai et al., 2007) or the dynamic clustering algorithm in Chen et al. (2004), extensive evaluations would be required, especially in scenarios with high velocity targets, to ensure the efficiency of the system. Tracking schemes with the ability of recovery from target loss can essentially be grouped as *fault-tolerant* or *reliable* protocols.

Yet another important direction of research is concerned with the MTT problem. For instance, the implications of tracking a number of targets moving in close proximity to each other on cluster or tree formation have not been investigated in the relevant schemes; instead, it is mostly assumed that targets maintain a sufficiently spaced apart condition and that each target is to be tracked by a dedicated cluster. Identity management schemes can be exploited to address the issue of targets with converging trajectories (Oh et al., 2006; Shin et al., 2003), albeit at the expense of significant computational overhead especially in the presence of a large number of objects.

The issue of MTT also needs to be extended beyond network level functions (e.g., coverage or data dissemination) to entail application level tasks as well. A promising direction in this area is tied up with the notion of *multiple mission assignment* first introduced in Rowaihy et al. (2007), and complemented in Bar-Noy et al. (2008) and Rowaihy et al. (2008). In this problem, it is required that a sensor node be assigned to at most one of the missions it is eligible for; whereas, it is considered legitimate for a mission to receive utility from multiple sensors. The problem of multiple mission assignment also gives rise to the issue of node reassignment; i.e., a node already assigned to a particular mission may later be required to be reassigned to another mission as it might be considered to be more helpful if it takes up the new post. Exploring the effects of such node reassignments on missions and the pertaining costs as well as the generalisation of multiple mission assignment through the investigation of the possibility or the implications of assigning a sensor to more than one mission puts forth an interesting area of research. Moreover, a network consisting of multi-modal sensors is able to detect different targets. In such networks, it might be desired to be able to activate the most appropriate modality in a particular sensor and then have it assigned to a mission depending on the priority of the object to be tracked. To the best of our knowledge, this issue has also not been approached before.

The optimisation of tracking protocols from the perspective of both the network structure as well as the application-specific requirements (e.g., MTT) is another significant topic of interest. To the best of our knowledge, the only method which leverages on optimisation techniques to set up a tree structure has been proposed in Lee et al. (2006), which is essentially a centralised algorithm with a heuristic initiative to reach near optimal solutions of the LP formulation of the problem. However, given the large overhead typical of centralised WSN algorithms, a promising direction would be the investigation of distributed schemes for achieving near optimal solutions, such as the iterative algorithms defined within the Network Utility Maximisation (NUM) framework (Chiang et al., 2007). In addition, such optimisations might require the mapping of quality of tracking criteria, defined at the application level, to lower level parameters which give the necessary foundation to measure the performance of the tracking protocols. Currently, we are working on defining specific quality of tracking measures, such as *robust detection* and *maximum target separability*, for the MTT problem. Similar ideas are also applicable for clustering techniques especially in the context of MTT or in designing tracking-specific MAC protocols (Swami et al., 2007).

Aside from layered optimisation, network-centric tracking protocols can also be designed for cross-optimisation with the other layers; for instance, as pointed out in Section 2.2, the joint optimisation of MAC and network-centric protocols is left as an open issue. Similarly, the application layer tracking protocols (e.g., CSP-based MMT algorithms) or even common services such as in-network data aggregation has the potential for the joint design and distributed implementation in companion with network-centric protocols.

A relatively unexplored area is the performance evaluation of OTSNs with precise mathematical models and with respect to quality of tracking parameters. The evaluation or comparison of tracking methods with different structures such as tree, cluster or leader has also not been undertaken yet. Quality parameters might encompass many aspects of the tracking problem; for instance: reliability of tracking in the presence of noisy measurements (Aslam et al., 2003), reliable delivery of a large number of measurements to the sink with minimal human intervention (Cinque et al., 2006), fault tolerance in detection and classification of targets (Clouqueur et al., 2004; Ding et al., 2005) and finally, extracting the target movement patterns for predicting the future target movements which comes in handy for reducing errors once combined with data mining methods (Tseng and Lu, 2009; Tseng and Lin, 2007) or Markov models (Peng et al., 2006).

As a final note, it might be worth mentioning that as the applications of OTSNs extend beyond military and industry, making impressions also on the crowded urban and natural environment scenarios, a new set of problems and/or issues is raised in this field, which in turn opens up new research opportunities for the interested community.

7 Concluding remarks

In this paper, the latest trends in mobile object tracking for WSNs have been investigated with the main emphasis put on the exploration of schemes from the communications-oriented and network-centric standpoints. We identified some tracking-centred design factors for consideration in both WSN services and in the other layers of the communication architecture as well as the parameters affecting the quality of tracking. We have categorised typical OTSN deployment scenarios with respect to the number of objects, type of the objects and modality of the sensors. Furthermore, the key modules comprising an object tracking system have been classified as either CSP-based algorithms or network-centric protocols. From the viewpoint of the WSN layered communication model, network-centric protocols may preferably be implemented within the network or the transport layers of the communication architecture.

The network-centric object tracking protocols, from the structural perspective, have been subsumed under the cluster-based, tree-based and leader-based categories. We presented a moderate overview and discussion of the existing schemes, pointed out their pros and cons, and also contrasted the approaches with reference to their competitive features. The major findings drawn from this comparison can be listed as follow:

- tree-based methods inflict larger overhead in contrast with cluster and leader-based methods
- tree- and leader-based schemes scale poorly in comparison with cluster-based methods, mainly on account of the costly expansion of their structures
- in order to inactivate unnecessary nodes, it is strongly recommended to either exploit regular structures or else make use of some prediction-based scheme.

Besides the issues and the shortcomings associated with the current techniques, we believe that an extensive menu of open research problems exists for object tracking in WSNs, some of which have been highlighted in this paper.

Acknowledgement

This work is supported in part by Iran Telecommunications Research Center (ITRC) under grant #1042/500.

References

Abdelzaher, T., Blum, B., Evans, D., George, J., George, S., Gu, L., He, T., Huang, C., Nagaraddi, P., Son, S., Sorokin, P. and Stankovic, J. (2004) 'A. wood, envirotrack: towards an environmental computing paradigm for distributed computing systems', *Proc. of ICDCS*, Tokyo, Japan, pp.582–589.

Abidi, B.R., Aragam, N.R., Yao, Y. and Abidi, M.A. (2008) 'Survey and analysis of multimodal sensor planning and integration for wide area surveillance', *ACM Computing Surveys*, Vol. 41, No. 1, December, pp.1–36.

Aghajarian, D. and Berangi, R. (2008) 'A fast distributed target tracking algorithm for low density binary sensor networks', *Proc. of the 2nd Int. Conf. on Sensor Technologies and Applications*, Cap Esterel, France, pp.1–6.

Akyildiz, I.F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002) 'A survey on sensor networks', *IEEE Communications Magazine*, pp.102–114.

Ali Alhmiedat, T. and Yang, S-H. (2007) 'A survey: localization and tracking mobile targets through wireless sensor network', *Proc. of PGNet*, Liverpool, England.

Amundson, I., Kushwaha, M., Kusy, B., Volgyesi, P., Simon, G., Koutsoukos, X. and Ledeczi, A. (2007) 'Time synchronization for multi-modal target tracking in heterogeneous sensor networks', *Workshop on Networked Distributed Systems for Sensing and Control*, Kalamata, Greece.

Arora, A., Dutta, P., Bapat, S., Kulathumani, V., Zhang, H., Naik, V., Mittal, V., Cao, H., Demirbas, M., Gouda, M., Choi, Y., Herman, T., Kulkarni, S., Arumugam, U., Nesterenko, M., Vora, A. and Miyashita, M. (2004) 'A line in the sand: a wireless sensor network for target detection, classification and tracking', *Computer Networks*, Vol. 46, pp.605–634.

Arora, A., Ramnath, R., Ertin, E., Sinha, P., Bapat, S., Naik, V., Kulathumani, V., Zhang, H., Cao, H., Sridharan, M., Kumar, S., Seddon, N., Anderson, C., Herman, T., Trivedi, N., Zhang, C., Nesterenko, M., Shah, R., Kulkarni, S., Aramugam, M., Wang, L., Gouda, M., Choi, Y., Culler, D., Dutta, P., Sharp, C., Tolle, G., Grimmer, M., Ferriera, B. and Parker, K. (2005) 'ExScal: elements of an extreme scale wireless sensor network', *11th IEEE Int. Conf. on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, August, Hong Kong, China, pp.102–108.

Aslam, J., Butler, Z., Constantin, F., Crespi, V., Cybenko, G. and Rus, D. (2003) 'Tracking a moving object with a binary sensor network', *Proc. of the ACM SenSys03*, Los Angeles, CA, pp.150–161.

Aurenhammer, F. (1991) 'Voronoi diagrams: a survey of a fundamental geometric data structure', *ACM Comput. Surv.*, Vol. 23, No. 3, pp.345–405.

Bar-Noy, A., Brown, T., Johnson, M.P., La Porta, T., Liu, O. and Rowaihy, H. (2008) 'Assigning sensors to missions with demands', *Proc. ALGOSENSORS'07*, Vol. 4837, Springer, Heidelberg, pp.114–125.

Bhatti, S. and Xu, J. (2009) 'Survey of target tracking protocols using wireless sensor networks', *5th Int. Conf. on Wireless and Mobile Communications (ICWMC)*, Cannes/La Bocca, France, pp.110–115.

Brooks, R., Ramanathan, P. and Sayeed, A.M. (2003) 'Distributed target classification and tracking in sensor networks', *Proc. of the IEEE, Special Issue on Sensor Networks*, Vol. 91, No. 8, pp.1163–1171.

Bruce, R.D., Jennings, J. and Ruc, D. (1997) 'Information invariants for distributed manipulation', *Int. J. of Robotics Research*, Vol. 16, No. 5, pp.673–702.

Cerpa, A., Elson, J., Estrin, D., Girod, L., Hamilton, M. and Zhao, J. (2001) 'Habitat monitoring: application driver for wireless communications technology', *First ACM SIGCOMM Workshop on Data Communication in Latin America and Caribbean*, San Jose, Costa Rica, pp.20–41.

Chakrabarty, K., Iyengar, S.S., Qi, H. and Cho, E. (2002) 'Grid coverage for surveillance and target location in distributed sensor networks', *IEEE Trans. on Computers*, Vol. 51, No. 12, pp.1448–1453.

Chang, W-R., Lin, H-T. and Cheng, Z-Z. (2008) 'CODA: a continuous object detection and tracking algorithm for wireless ad hoc sensor networks', *Proc. of the 5th IEEE CCNC*, Las Vegas, NV, pp.168–174.

Chen, L., Cetin, M. and Willsky, A.S. (2005) 'Distributed data association for multi-target tracking in sensor networks', *Proc. of Int. Conf. on Information Fusion*, Philadelphia, Pennsylvania.

Chen, W-P., Hou, J.C. and Sha, L. (2004) 'Dynamic clustering for acoustic target tracking in wireless sensor networks', *IEEE Trans. on Mobile Computing*, Vol. 3, No. 3, pp.258–271.

Chen, Y-S. and Ann, S-Y. (2005) 'VE-Mobicast: a variant-egg-based mobicast routing protocol for sensornets', *Proc. of the 40th IEEE ICC 2005*, Seoul, Korea. Vol. 5, pp.3020–3024.

Chiang, M., Low, S.H., Calderbank, A.R. and Doyle, J.C. (2007) 'Layering as optimization decomposition: a mathematical theory of network architectures', *Proceedings of the IEEE*, Vol. 95, No. 1, January, pp.255–312.

Chong, C-Y., Zhao, F., Mori, S. and Kumar, S. (2003) 'Distributed tracking in wireless sensor networks', *Proc. of the 6th Int. Conf. on Information Fusion*, Cairns, Queensland, Australia, pp.431–438.

Chu, M., Haussecker, H. and Zhao, F. (2002) 'Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks', *Int. J. of High Performance Comput. Appl.*, Vol. 16, No. 3, pp.293–313.

Cinque, M., Cotroneo, D., De Caro, G. and Pelella, M. (2006) 'Reliability requirements of wireless sensor networks for dynamic structural monitoring', *Proc. of Int. Conf. on Dependable Systems and Networks (DSN)*, Philadelphia, PA, USA.

Clouqueur, T., Saluja, K.K. and Ramanathan, P. (2004) 'Fault tolerance in collaborative sensor networks for target detection', *IEEE Trans. on Computers*, Vol. 53, No. 3, pp.320–333.

Cormen, T.H., Leiserson, C.E. and Rivest, R.L. (1990) *Introduction to Algorithms*, McGraw-Hill, New York.

Demirbas, M. and Lu, X. (2007) 'Distributed quad-tree for spatial querying in wireless sensor networks', *Proc. of ICC*, Glasgow, Scotland, pp.3325–3332.

Demirbas, M., Arora, A. and Kulathumani, V. (2006) 'GLANCE: a lightweight querying service for wireless sensor networks', *Proc. of OPODIS*, Bordeaux, France, pp.242–257.

Ding, M., Chen, D., Thaeler, A. and Cheng, X. (2005) 'Fault-tolerant target detection in sensor networks', *Proc. of the IEEE WCNC*, pp.2362–2368.

Doucet, A., deFreitas, N. and Gordon, N. (2001) *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York.

Fan, L., Wang, H. and Wang, H. (2006) 'A solution of multi-target tracking based on FCM Algorithm in WSN', *Proc. of the 4th Annual IEEE Int. Conf. on Pervasive Computing and Communications Workshops (PERCOMW)*, Pisa, Italy, pp.290–294.

Fasolo, E., Rossi, M., Widmer, J. and Zorzi, M. (2007) 'In-network aggregation techniques for wireless sensor networks: a survey', *IEEE Wireless Comm. Mag.*, Vol. 14, No. 2, pp.70–87.

Felemban, E., Lee, C-G. and Ekici, E. (2006) 'MMSPEED: multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks', *IEEE Trans. on Mobile Comp.*, Vol. 5, No. 6, pp.738–754.

Fuemmeler, J.A. and Veeravalli, V.V. (2008) 'Smart sleeping policies for energy effiicent tracking in sensor networks', *IEEE Trans. on Sig. Proc.*, Vol. 56, No. 5, pp.2091–2101.

Goel, S. and Imielinski, T. (2001) 'Prediction-based monitoring in sensor networks: taking lessons from MPEG', *ACM SIGCOM, Comp. Comm. Review*, Vol. 31, No. 5, pp.82–98.

Gui, C. and Mohapatra, P. (2004) 'Power conservation and quality of surveillance in target tracking sensor networks', *Proc. of the ACM MobiCom'04*, pp.129–143.

Gui, C. and Mohapatra, P. (2005) 'Virtual patrol: a new power conservation design for surveillance using sensor networks', *4th Int. Symp. on Information Processing in Sensor Networks (IPSN)*, Los Angeles, CA, USA, pp.246–253.

Gurses, E., Lin, Y. and Boutaba, R. (2009) 'Distributed quality-lifetime maximization in wireless video sensor networks', *Proc. of IEEE ICC*, Dresden, Germany, pp.1–6.

Hall, D.L. and Llinas, J. (2001) *Handbook of Multisensor Data Fusion*, CRC Press, Boca Raton, Florida, USA.

He, T., Krishnamurthy, S., Luo, L., Yan, T., Gu, L., Stoleru, R., Zhou, G., Cao, Q., Vicaire, P., Stankovic, J.A., Abdelzaher, T.F., Hui, J. and Krogh, B.H. (2006) 'VigilNet: an integrated sensor network system for energy efficient surveillance', *ACM Trans. on Sensor Networks (TOSN)*, Vol. 2, No. 1., pp.1–38.

He, T., Stankovic, J.A., Lu, C. and Abdelzaher, T. (2003) 'SPEED: a stateless protocol for real-time communication in sensor networks', *Proc. of ICDCS*, Providence, RI, USA, pp.46–55.

Heinzelman, W.B., Chandrakasan, A.P. and Balakrishnan, H. (2002) 'An application-specific protocol architecture for wireless microsensor networks', *IEEE Trans. on Wireless Comm.*, Vol. 1, No. 4, pp.660–670.

Huang, Q., Lu, C. and Roman, G-C. (2004) 'Design and analysis of spatiotemporal multicast protocols for wireless sensor networks', *Telecommunication Systems, Special Issue on Wireless Sensor Networks*, Vol. 26, pp.129–160.

Huang, X. and Fang, Y. (2007) 'Multiconstrained QoS multipath routing in wireless sensor networks', *Wireless Networks*, Springer, Vol. 14, No. 4, August, pp.465–478.

Huiyu, Z., Taj, M. and Cavallaro, A. (2008) 'Target detection and tracking with heterogeneous sensors', *IEEE Journal of Selected Topics in Signal Processing*, Vol. 2, No. 4, August, pp.503–513.

Hwang, S-F., Lu, K-H., Yang, L-R. and Dow, C-R. (2008) 'Efficient data reporting for object tracking in wireless sensor networks with mobile sinks', *Proc. of the 14th IEEEAsia-Pacific Conf. on Comm. (APCC)*, Tokyo, Japan, pp.1–5.

Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J. and Silva, F. (2003) 'Directed diffusion for wireless sensor networking', *IEEE/ACM Trans. on Networking*, Vol. 11, No. 1, pp.2–16.

Ji, X., Zha, H., Metzner, J.J. and Kesidis, G. (2004) 'Dynamic cluster structure for object detection and tracking in wireless ad-hoc sensor networks', *Proc. of the IEEE ICC 2004*, pp.3807–3811.

Jiang, B., Han, K., Ravindran, B. and Cho, H. (2008) 'Energy efficient sleep scheduling based on moving directions in target tracking sensor networks', *Proc. of IEEE IPDPS*, Miami, Florida, USA, pp.1–10.

Jin, G-Y., Lu, X-Y. and Park, M-S. (2006) 'Dynamic clustering for object tracking in wireless sensor networks', *Ubiquitous Computing Systems (UCS)*, LNCS, 2006, Vol. 4239, pp.200–209.

Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L-S. and Rubenstein, D. (2002) 'Energy-efficient computing for wildlife tracking: design tradeoffs and early experience with zebranet', *Proc. of the 10th Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, October, San Jose, CA, USA, pp.96–107.

Kim, W., Mechitov, K., Choi, J-Y. and Ham, S. (2005) 'On target tracking with binary proximity sensors', *Proc. of the IEEE IPSN*, Los Angeles, California, USA, pp.301–308.

Kim, Y-W., Lee, J-S., Yoo, S-K. and Kim, H-J. (2007) 'An ID-based USN management architecture model', *Proc. of the 9th Int. Conf. on Advanced Communication Technology, 2007*, Phoenix Park, Korea, Vol. 1, pp.251–254.

Kulathumani, V., Arora, A., Demirbas, M. and Sridharan, M. (2007) 'Trail: a distance sensitive WSN service for distributed object tracking', *Proc. of EWSN*, Delft, The Netherlands.

Kulathumani, V., Arora, A., Sridharan, M. and Demirbas, M. (2009) 'Trail: a distance-sensitive sensor network service for distributed object tracking', *ACM Trans. on Sensor Networks*, Vol. 5, No. 2, March, pp.1–40.

Kumar, S., Alaettinglu, C. and Estrin, D. (2000) 'SCalable object-tracking through unattended techniques (SCOUT)', *Int. Conf. on Network Protocols (ICNP)*, November, Osaka, Japan, pp.253–262.

Kung, H.T. and Vlah, D. (2003) 'Efficient location tracking using sensor networks', *Proc. of the IEEE WCNC*, New Orleans, Louisiana, USA, pp.1954–1961.

Kushwaha, M., Amundson, I., Volgyesi, P., Ahammad, P., Simon, G., Koutsoukos, X., Ledeczi, A. and Sastry, S. (2008) 'Multi-modal target tracking using heterogeneous sensor networks', *17th Int. Conf. on Computer Communications and Networks (ICCCN)*, August, St. Thomas, US Virgin Islands.

Lee, C-T., Lin, F.Y-S. and Wen, Y-F. (2006) 'An efficient object tracking algorithm in wireless sensor networks', *Proc. of the 9th Joint Conf. on Information Sciences (JCIS)*, Kaohsiung, Taiwan.

Li, D., Wong, K.D., Hu, Y.H. and Sayeed, A.M. (2002) 'Detection, classification and tracking of targets', *IEEE Signal Proc. Mag.*, Vol. 19, No. 2, pp.17–29.

Lin, C-Y., Peng, W-C. and Tseng, Y-C. (2006) 'Efficient in-network moving object tracking in wireless sensor networks', *IEEE Trans. on Mobile Computing*, Vol. 5, No. 8, pp.1044–1056.

Lin, J., Xiao, W., Lewis, F.L. and Xie, L. (2009) 'Energy-efficient distributed adaptive multisensor scheduling for target tracking in wireless sensor networks', *IEEE Trans. on Instrumentation and Measurement*, Vol. 58, No. 6, pp.1886–1896.

Lin, Y-B. and Hwang, S-Y. (1996) 'Comparing the PCS location tracking strategies', *IEEE Trans. on Vehicular Technologies*, Vol. 45, No. 1, pp.114–121.

Liu, B-H., Ke, W-C., Tsai, C-H. and Tsai, M-J. (2008) 'Constructing a message-pruning tree with minimum cost for tracking moving objects in wireless sensor networks is NP-complete and an enhanced data aggregation structure', *IEEE Trans. on Computers*, Vol. 57, No. 6, June, pp.849–863.

Liu, J., Chu, M. and Reich, J.E. (2007) 'Multitarget tracking in distributed sensor networks', *IEEE Signal Proc. Mag.*, Vol. 24, No. 3, pp.36–46.

Liu, J., Liu, J., Reich, J., Cheung, P. and Zhao, F. (2003) 'Distributed group management for track initiation and maintenance in target localization applications', *Proc. of the 2nd IPSN Workshop*, Palo Alto, CA, USA, pp.113–128.

Liu, J., Zhao, F. and Petrovic, D. (2005) 'Information-directed routing in ad hoc sensor networks', *IEEE J. on Sel. Areas in Comm.*, Vol. 23, No. 4, pp.851–860.

Maheswararajah, S., Halgamuge, S.K. and Premaratne, M. (2009) 'Sensor scheduling for target tracking by suboptimal algorithms', *IEEE Trans. on Vehicular Technology*, Vol. 58, No. 3, pp.1467–1479.

Mechitov, K., Sundresh, S., Kwon, Y. and Agha, G. (2003) 'Cooperative tracking with binary-detection sensor networks', *Proc. of the ACM SenSys*, Los Angeles, CA, USA, pp.332–333.

Monari, E., Voth, S. and Kroschel, K. (2008) 'An object- and task-oriented architecture for automated video surveillance in distributed sensor networks', *Proc. of the IEEE 5th Int. Conf. on Advanced Video and Signal Based Surveillance*, New Mexico, USA, pp.339–346.

Nebot, E., Masson, F., Guivant, J. and Durrant-Whyte, H. (2002) *Robust Simultaneous Localization and Mapping for Very Large Outdoor Environment, Experimental Robotics VIII*, Springer, pp.200–209.

Ochirsuren, E., Indrusiak, L.S. and Glesner, M. (2008) 'An Actor-oriented group mobility model for wireless ad hoc sensor networks', *Int. Conf. on Distributed Computing Systems (ICDCS)*, June, Beijing, China, pp.174–179.

Oh, S., Hwang, I., Roy, K. and Sastry, S. (2005) 'A fully automated distributed multiple-target tracking and identity management algorithm', *Proc. of AIAA Guidance, Navigaiton and Contorl Conf.*, San Francisco, California, USA.

Oh, S., Russell, S. and Sastry, S. (2004) 'Markov chain Monte Carlo data association for general multiple-target tracking problems', *Proc. of the 43rd IEEE Conf. on Decision and Control*, Nassau, Bahamas, Vol. 1, pp.735–742.

Olule, E., Wang, G., Guo, M. and Dong, M. (2007) 'RARE: an energy-efficient target tracking protocol for wireless sensor networks', *Proc. of the Int. Conf. on Parallel Processing Workshop (ICPPW)*, Xian, China, pp.76–81.

Orguner, U. and Gustafsson, F. (2009) 'Distributed target tracking with propagation delayed measurements', *Proc. Of the 12th Int. Conf. on Information Fusion*, Seattle, Washington, USA, pp.802–809.

Pahalawatta, P.V., Pappas, T.N. and Katsaggelos, A.K. (2004) 'Optimal sensor selection for video-based target tracking in a wireless sensor network', *2004 Int. Conf. on Image Processing (ICIP)*, Singapore, pp.3073–3076.

Park, S. (2006) *Performance Analysis of Data Aggregation Schemes for Wireless Sensor Networks*, PhD Thesis, North Carolina State University, USA.

Parker, L.E. (1997) 'Cooperative motion control for multi-target observation', *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems*, Grenoble, France, pp.1591–1597.

Pashazadeh, S. and Sharifi, M. (2007) 'A comparative study of two typical fusion methods for target tracking in wireless sensor networks', *Proc. of the 3rd Int. Conf. on Information and Knowledge Technology (IKT)*, Mashhad, Iran.

Pattem, S., Poduri, S. and Krishnamachari, B. (2003) 'Energy-quality tradeoff for target tracking in wireless sensor networks', *Proc. of IPSN 2003*, Palo Alto, CA, USA, pp.32–46.

Peng, W-C., Ko, Y-Z. and Lee, W-C. (2006) 'On mining moving patterns for object tracking sensor networks', *Proc. of the 7th Int. Conf. on Mobile Data Management (MDM)*, Nara, Japan, pp.41–44.

Rowaihy, H., Eswaran, S., Johnson, M., Verma, D., Bar-Noy, A., Brown, T. and La Porta, T. (2007) 'A survey of sensor selection schemes in wireless sensor networks', *Proc. of Defense and Security Symposium on Unattended Ground, Sea and Air Sensor Technologies and Applications IX*, Orlando, FL, USA.

Rowaihy, H., Johnson, M., Bar-Noy, A., Brown, T. and La Porta, T. (2008) 'Assigning sensors to competing missions', *Proc. of IEEE GLOBECOM*, New Orleans, LA, USA, pp.1–6.

Shin, J., Guibas, L.J. and Zhao, F. (2003) 'A distributed algorithm for managing multi-target identities in wireless ad hoc sensor networks, *Proc. of IPSN 2003, LNCS*, Vol. 2634, Springer, Heidelberg, pp.223–238.

Shirvastava, N., Mudumbai, R., Madhow, U. and Suri, S. (2006) 'Target tracking with binary proximity sensors: fundamental limits, minimal descriptions and algorithms', *Proc. of the ACM SenSys*, Boulder, Colorado, USA, pp.251–264.

Skibniewski, M.J. and Jang, W-S. (2006) 'Ubiquitous computing: object tracking and monitoring in construction processes utilizing zigbee networks', *Proc. of the 23rd Int. Symp. on Automation and Robotics in Construction (ISARC 2006)*, Tokyo, Japan, pp.1–6.

Smith, D. and Singh, S. (2006) 'Approaches to multisensor data fusion in target tracking: a survey', *IEEE Trans. on Knowledge and Data Engineering*, Vol. 18, No. 12, pp.1696–1710.

Song, L. and Hatzinakos, D. (2007) 'A cross-layer architecture of wireless sensor networks for target tracking', *IEEE/ACM Trans. on Networking*, Vol. 15, No. 1, pp.145–158.

Soro, S. and Heinzelman, W.B. (2005) 'On the coverage problem in video-based wireless sensor networks', *Proc. of the 2nd Workshop on Broadband Advanced Sensor Networks (BaseNets)*, Boston, MA, USA, Vol. 2, pp.932–939.

Stark, D. and Davis, J. (2004) 'Friendly object tracking and foreign object detection and localization with an SDAC wireless sensor network', *10th IEEE Int. Workshop on Future Trends of Distributed Computing Systems (FTDCS)*, May, Suzhou, China, pp.30–36.

Sung, Y., Misra, S., Tong, L. and Ephremides, A. (2007) 'Cooperative routing for distributed detection in large sensor networks', *IEEE J. on Sel. Areas in Comm.*, Vol. 25, No. 2, pp.471–483.

Swami, A., Zhao, Q., Hong, Y-W. and Tong, L. (2007) *Wireless Sensor Networks, Signal Processing and Communications Perspectives*, John Wiley & Sons, England.

Tran, S.P.M. and Yang, T.A. (2006) 'OCO: optimized communication and organization for target tracking in wireless sensor networks', *Proc. of the IEEE Int. Conf. on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC)*, Taichung, Taiwan, pp.428–435.

Tsai, H-W., Chu, C-P. and Chen, T-S. (2007) 'Dynamic object tracking in wireless sensor networks, *Computer Comm.*, Vol. 30, pp.1811–1825.

Tseng, V.S. and Lin, K.W. (2007) 'Energy efficient strategies for object tracking in sensor networks: a data mining approach', *The J. of Systems and Software*, Vol. 80, No. 10, pp.1678–1698.

Tseng, V.S. and Lu, E.H-C. (2009) 'Energy-efficient real-time object tracking in multi-level sensor networks by mining and predicting movement patterns', *J. of Systems and Software*, Vol. 82, No. 4, pp.697–706.

Tseng, Y-C., Kuo, S-P., Lee, H-W. and Huang, C-F. (2004) 'Location tracking in a wireless sensor network by mobile agents and its data fusion strategies', *The Computer Journal*, Vol. 47, No. 4, pp.448–460.

Vuran, M.C., Akan, O.B. and Akyildiz, I.F. (2004) 'Spatio-temporal correlation: theory and applications for wireless sensor networks', *Computer Networks*, Vol. 45, No. 3, June, pp.245–259.

Walchli, M., Skoczylas, P., Meer, M. and Braun, T. (2007) 'Distributed event localization and tracking with wireless sensors', *Proc. of the 5th Int. Conf. on Wired/Wireless Internet Communications*, Coimbra, Portugal, pp.247–258.

Wang, W., Srinivasan, V., Wang, B. and Chua, K-C. (2008) 'Coverage for target localization in wireless sensor networks', *IEEE Trans. on Wireless Comm.*, Vol. 7, No. 2, pp.667–676.

Wang, Y., Yang, J., Yang, X. and Zhang, D. (2008) 'MTT-mobicast: maneuvering target tracking mobicast protocol for wireless ad hoc sensor networks', *Proc. of the 8th IEEE CIT*, Sydney, Australia, pp.821–826.

Wang, Z., Lou, W., Wang, Z., Ma, J. and Chen, H. (2010) 'A novel mobility management scheme for target tracking in cluster-based sensor networks', *Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)*, Santa Barbara, California, USA, pp.172–186.

Xu, Y., Heidemann, J. and Estrin, D. (2001) 'Geography informed energy conservation for ad hoc routing', *Proc. of ACMMobile Computing and Networking*, Rome, Italy, pp.70–84.

Xu, Y., Winter, J. and Lee, W-C. (2004) 'Dual prediction-based reporting for object tracking sensor networks', *Proc. of the 1st MobiQuitous Ann. Conf.*, Boston, MA, USA, pp.154–163.

Xu, Y., Winter, J. and Lee, W-C. (2004) 'Prediction-based strategies for energy saving in object tracking sensor networks', *Proc. of the 2004 IEEE MDM Int. Conf.*, Berkeley, CA, USA, pp.346–357.

Yang, H. and Sikdar, B. (2003) 'A protocol for tracking mobile targets using sensor networks', *Proc. of the 1st IEEE Int. Workshop on Sensor Network Protocols and Applications*, Anchorage, Alaska, pp.71–81.

Yarvis, M., Kushalnagar, N., Singh, H., Rangarajan, A., Liu, Y. and Singh, S. (2005) 'Exploiting heterogeneity in sensor networks', *IEEE INFOCOM*, Miami, FL, USA, Vol. 2, pp.878–890.

Yen, L-H. and Yang, C-C. (2006) 'Mobility profiling using Markov chains for tree-based object tracking in wireless sensor networks', *IEEE Int. Conf. on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC)*, Taichung, Taiwan, Vol. 2, pp.220–225.

Yick, J., Mukherjee, B. and Ghosal, D. (2008) 'Wireless sensor network survey', *Computer Networks*, Vol. 52, pp.2292–2330.

Yu, X., Niyogi, K., Mehrotra, S. and Venkatasubramanian, N. (2004) 'Adaptive target tracking in sensor networks', *Proc. of the 2004 Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS'04)*, San Diego, CA, USA.

Zhang, C., Wu, J. and Tu, G. (2006) 'Object tracking and QoS control using infrared sensor and video camera', *Proc. of the IEEE Int. Conf. on Networking, Sensing and Control (ICNSC)*, Ft. Lauderdale, Florida, USA, pp.974–979.

Zhang, W. and Cao, G. (2004) 'DCTC: dynamic convoy tree-based collaboration for target tracking in sensor networks', *IEEE Tran. on Wireless Comm.*, Vol. 3, No. 5, pp.1689–1701.

Zhang, W. and Cao, G. (2004) 'Optimizing tree reconfiguration for mobile target tracking in sensor networks', *Proc. of the IEEE INFOCOM*, Vol. 4, pp.2434–2445.

Zhao, F., Shin, J. and Reich, J. (2002) 'Information-driven dynamic sensor collaboration', *IEEE Signal Proc. Mag.*, Vol. 19, No. 2, pp.61–72.

Note

[1] A *signature* packet contains detailed signature information which can be either raw data or the extracted feature of a signal.

Bibliography

Akkaya, K. and Younis, M. (2005) 'A survey on routing protocols for wireless sensor', *Ad hoc Networks*, Vol. 3, No. 3, pp.325–349.

Cheng, Y., Yang, L. and Li, Q. (2006) 'Mining frequent moving patterns of objects in sensor networks', *Proc. of the 2006 IEEE Int. Conf. on Information Acquisition*, Weihai, China, pp.797–801.

Fan, K-W. and Sinha, P. (2008) 'Distributed online data aggregation for large scale sensor networks', *Proc. of 5th IEEE MASS*, Atlanta, Georgia, pp.153–162.

Fan, K-W., Liu, S. and Sinha, P. (2007) 'Structure-free data aggregation in sensor networks', *IEEE Trans. on Mobile Computing*, Vol. 6, No. 8, pp.929–942.

Yang, W., Fu, Z., Kim, J. and Park, M-S. (2007) 'An adaptive dynamic cluster-based protocol for target tracking in wireless sensor networks', *Proc. of the joint 9th Asia-Pacific Web and 8th Int. Conf. on Web-Age Information Management (APWeb/WAIM)*, LNCS, Vol. 4505, pp.157–167.