ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

DOT: Dynamical Optimal Target Tracking For Additive and Multiplicative Reduction Using Sensors

M. Poojitha, Bullarao Domathoti, P.Nageswara Rao

M. Tech Student, Dept. of CSE., SITS, JNTUA University, Tirupati , India.

Assistant Professor, Dept. of. CSE, SITS, JNTUA University, Tirupati , India.

Assistant Professor, Dept. of. CSE, SITS, JNTUA University, Tirupati , India.

ABSTRACT: The use of wireless sensor networks (WSN) in tracking applications is growing at a fast pace. In these applications, the sensor nodes discover, monitor and track an event or target object. A significant number of proposals relating the use of WSNs for target tracking have been published to date. However, they either focus on the tracking algorithm or on the communication protocol, and none of them address the problem integrally. In this paper, a comprehensive proposal for target detection and tracking is discussed. We introduce a tracking algorithm to detect and estimate a target location. Moreover, we introduce a low-overhead routing protocol to be used along with our tracking algorithm. The proposed algorithm has low computational complexity and has been designed considering the use of a mobile sink while generating minimal delay and packet loss. We also discuss the results of the evaluation of the proposed algorithms. reduction on combinations of task sensors; and efficient search of optimal sensor motion. The performance of the proposed coordination strategy is illustrated by simulations.

KEYWORDS: wireless sensor network; intruder tracking and detection; mobile sink;

I. INTRODUCTION

Wireless sensor networks (WSN) are comprised of many small devices, called motes, which are deployed over an area of interest in order to detect and monitor events, or to track persons or objects as they move through the sensed area [1]. In surveillance and tracking applications, nodes in a WSN work together to monitor the existence of intruding targets, such as persons or vehicles [2]. Since motes have limited resources, particularly computer power and energy supply, common goals of these applications are reliable detection of targets and fast event notifications with minimal resources consumption.A WSN monitoring application must periodically collect sensed data and use it to reconstruct the overall status of the monitored area through data aggregation. For this purpose, when the sensor nodes detect an event, they record it and rely on a distributed routing protocol to send the relevant information towards a base station or sink. The sink is a device with much more resources in charge of collecting the information received from the sensor nodes, processing it, and, if necessary, taking the appropriate actions.

It is important that the target events are detected with an acceptable degree of accuracy. To achieve this goal, different algorithms could be used to estimate the intruder location within the sensed area. In addition, the event report should be received by the sink in a very short time, especially for time-critical events. The connectivity issues related to the transmission range of nodes should be under control, and the selected communication protocols must provide a minimum delay and packet loss.

In recent years the research community has proposed several solutions for target tracking in WSNs, focusing mostly on specific elements of the architecture, such as the tracking algorithm or the communications protocol, lacking a holistic solution to the problem. To address this limitation, this paper combines a communication architecture with a tracking algorithm to achieve an efficient intruder detection and tracking system.The algorithm assumes the use of binary sensors, a mobile sink, and the IEEE 802.15.4 standard for radio communications. We also introduce the

Mobile-Sink Routing for Large Grids (MRLG) algorithm, intended to support sink mobility for time-critical applications, for use along with our tracking algorithm. The MRLG routing protocol allows reducing the routing load by relying on local route recovery processes, which provides significant efficiency in scenarios with a large number of sensor nodes. Both algorithms have a low computational complexity.

Many applications may benefit from the use of a mobile sink, among which are target tracking and intrusion detection applications. When an intruder is detected, the sensor nodes report an alarm to the mobile sink, which monitors the progression of the intruder and takes appropriate actions, such as sending the intruder location to the security personnel. The security personnel, endowed with a mobile terminal (sink), may move towards the intruder in an attempt to seize him, using the location feedback provided by the WSN. On the other hand, it has been shown that mobile sinks may improve the WSN lifetime by spreading the transmission overhead of nodes that are close to the sink [3]. However, the main challenge introduced by sink mobility is that the WSN must be continuously and quickly reconfigured to adapt to the topology changes, making sure that information loss is minimal.

II. TRACKING ALGORITHMS

With respect to the application layer, various approaches have been used in target tracking algorithms. Some are based on different types of measurements, such as the received signal strength (RSS) [5], angle of arrival (AOA) [5,6], time of arrival (TOA) [7], time difference of arrival (TDOA) [8], extended Kalman filters (EKF) [9], and hybrid approaches [10,11], to mention a few. Some others are based on the use of binary sensors, which provide only 1-bit information regarding the presence or absence of a target in their detection area [12].Concerning proposals using a WSN for target tracking purposes, Li et al. [13] addressed the topic of detection and tracking of a single target in a WSN with a static sink, using the coordination between routing protocols and location algorithms. They then extended it to multiple target tracking. In [14], the authors analyzed the fundamental performance limits of tracking a target in a two-dimensional field of binary proximity sensors, determining the accuracy with which a target's trajectory can be tracked.

In a later work [15] they extended it to a multiple target tracking solution; in their work they do not address communication protocols used by the sensor nodes, focusing solely on the efficacy of collaborative tracking. Arora et al. in [16] studied target tracking and classification. They defined the specifications of WSN deployment for target detection, identifying the best types of sensors to be used by these applications in dense network environments, when focusing on human targets or vehicles. Sheng and Hu proposed a maximum likelihood acoustic source location estimation method for target localization in [17]. However, since this method uses nonlinear optimization, it is difficult to obtain closed form solutions. Chen et al. [18] developed an application for control and surveillance in large-scale, real-time WSNs, using a multi-target tracking algorithm, which combined a multisensor fusion method and a Markov chain Monte Carlo Data Association (MCMCDA) algorithm. Their solution is able to automatically start and finish the tracking procedure. The Continuous Object Detection and tracking Algorithm (CODA) was proposed in [19] by Chang et al. to detect and track the spread of continuous objects such as wild fires, toxic gases, etc. Kim et al. introduced a target tracking algorithm based on binary sensors and a static sink. The algorithm utilizes the sensor outputs to estimate individual positions in the path of the target and finds the trajectory that best fits the path points. This estimated trajectory is then used to estimate the current position of the target. Wang et al. presented in [11] an approach for target tracking for WSN by combining maximum likelihood estimation and Kalman filtering using the distance measurement. The maximum likelihood estimator is used for prelocalization of the target and measurement conversion. The converted measurement and its associated noise statistics are then used in a standard Kalman filter for recursive update of the target state.

III. LITERATURE SURVEY

Concerning solutions supporting mobile sinks, Tsaia et al. [20] proposed a Dynamical Object Tracking (DOT) algorithm, devised to be used by a mobile source (sink) to chase a moving target. The algorithm uses the knowledge of spatial neighborhood defined on a planar graph, where the face neighbors are identified by a Gabriel Graph. When a target is detected, the sensors need to record the target tracks. When the source requires the target location, it sends a query to the node keeping the track information, which replies with the tracking information, until the source catches

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

the target. Zhang et al. [21] proposed a distributed management scheme that uses a set of access points to support transmission from a sensor to the sink. To update the location information, the sink was allowed to choose from different access points while in motion, so that source nodes can be informed about the sink location at all times.

Since energy efficiency is a critical issue in WSNs, several authors have addressed energy-efficient tracking solutions. He et al. proposed in [22] a monitoring system for use in military applications, such as a surveillance system, that is able to operate for long periods of time. The system, designed from the standpoint of energy efficiency, was evaluated using a network equipped with 70 MICA2 motes with dual-axis-magnetometers. He et al. later developed VigilNet [23], a large-scale real-time WSN system that allows detecting, tracking and classifying targets within a reasonable time, while making efficient use of energy. VigilNet is a system designed for spontaneous military operations in remote areas, where events of interest happen infrequently and with a short duration, such as intruder related events. The VigilNet network infrastructure is based on multi-path diffusion tree rooted at bases. In [12], Cao et al. [24] established the relationship between the system parameters and attributes of surveillance, applied to both fixed and moving targets. The authors adopted the model of duty-cycle planning for individual, unsynchronized nodes, allowing nodes to periodically sleep and wake up. In [9], Lin et al. introduced an EKF-based distributed adaptive multisensor scheduling scheme for energy efficiency, to improve tracking accuracy. The sensor scheduling problem is formulated as an optimization problem and solved by a sequential three-step heuristic algorithm.

Our proposed tracking algorithm differs from the former ones for several reasons. Since we adopt binary detection sensors to achieve a low-cost solution, we propose a new tracking algorithm based on information aggregation that specifically targets this type of sensors. Moreover, since we want to support mobile sinks to model intruder pursuit by the security personnel, we required a robust and efficient algorithm that quickly adapts to topology changes. To achieve this goal, we devised an algorithm with low computational complexity. Additionally, the previously reviewed proposals are focused on the application layer and do not consider the routing protocol to be used. In contrast, the proposed tracking algorithm relies on our Mobile-sink Routing for Large Grids routing protocol, which provides high efficiency and very low routing overhead in scenarios with a large number of sensors and a mobile sink.

IV. ROUTING PROTOCOLS

Many of the proposed routing protocols that assume sink mobility have been devised for energy efficiency, regardless of the transmission delay [25]. Tong et al. proposed the SEnsor Networks with Mobile Agents (SENMA) in [26], a network architecture for low power and large scale sensor networks. In SENMA, mobile agents or sinks are the only receiving terminals in data collections. Thus, when a node detects an event, it must wait for a mobile agent to be in its transmission range to send the collected data, avoiding the need of multi-hop transmission. The Scalable Energy-Efficient Asynchronous Dissemination (SEAD) protocol was introduced in [27] by Kim et al., which assigns specific fixed nodes as the sink's access nodes. The access node is used to represent the moving sink. Sensor nodes deliver data to the access node, which in turn delivers it to the sink without exporting the sink's location information to the rest of the nodes. The protocol assumes that each sensor node is aware of its own geographic location.

MobiRoute [28] is a routing protocol that focuses on scenarios where all the sensor nodes are fixed and have limited energy. Sink mobility is used to increase the network lifetime. The MULE architecture [29] also exploits mobility for energy efficient non-real-time data collection in sparse sensor networks. Song and Hatzinakos proposed the Transmission Scheduling Algorithm Sensor Networks with Mobile Sink (TSA-MSSN) for WSN-based applications with large latency tolerance, minimizing the energy consumption costs. More recently, Rao and Biswas [30] studied the impact of different data collection modes on energy, including network assisted sink navigation.

Concerning those proposals that study sink mobility combined with the IEEE 802.15.4 standard, Chen and Ma [31] conducted a performance evaluation considering mobile sinks and using the AODV routing protocol without major modifications. They considered mainly the delay-energy metric for assessing the performance of different data collection schemes. Vlajic et al. [32] presented different strategies for reducing the mobility-related overhead in 802.15.4/ZigBee-based WSNs, assessing the effectiveness of different path-constrained mobile sink trajectories and their suitability in real environments.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

Our MRLG algorithm differs from these proposals since for target tracking applications, it is required that the sink receives near real-time feedback about the intruder position in order to pursue him (during short periods of time). Thus, the sink trajectory cannot be constrained. Besides, it does not aim at meeting minimum energy consumption requirements and is thus outside the family of routing protocols that help extending the WSN lifetime.Recently, the Collection Tree Protocol (CTP) [33] has been proposed for WSNs. CTP is a sink-announcement based protocol where sink announcements are fully propagated throughout the WSN to reach all nodes, allowing the creation of a route tree from all the sensor nodes towards the sink. Any sensor node willing to send or forward a report packet to the sink will consult its routing table to see if there is a valid path towards the sink, and then send the information using that route. In our experiments we compared MRLG against the CTP routing protocol to assess performance gains.

V. Intruder Tracking Algorithm

In intruder detection and tracking applications, a real-time system is actually a soft real-time system, where some latency is allowed [36]. Identifying emergency situations in a few seconds will suffice for satisfying the application requirements. Therefore, applications for intruder detection and tracking require low end-to-end delay values for sensor reports. In the scope of intruder tracking, the challenge is to ensure that the delays from different sources are small enough to allow a timely data fusion, and that the data fusion algorithm is able to provide intruder position estimations as quickly as possible. To meet this goal efficiently, we should have: (i) routing algorithms that are robust, and that quickly adapt to topology changes, while imposing a low routing overhead; and (ii) data fusion algorithms that are able to track the intruder using a small number of reports, avoiding the use of large time windows for report gathering, and also being flexible in the presence of packet loss.

In our approach, the following additional issues should also be taken into consideration: The sensor detection is binary within a range of 10 m (equal to the minimum distance between sensors), and sensor coverage is omnidirectional. This implies that, for the proposed grid deployment strategy, the number of sensors simultaneously detecting an intruder can range from one to four.

The intruder location is constantly updated by the sink node based on the different sensors reports received, and taking into account the timestamp associated with each individual report id.

Algorithm (1) summarizes the process used to continuously estimate the intruder location. This algorithm adopts a report grouping strategy for data fusion purposes that is able to reliably estimate the location of a quickly moving intruder while introducing a low degree of complexity and few calculations. This algorithm was optimized to achieve near real-time position updating for single-target tracking, which was made possible through low computational overhead and incremental filtering based on small time-intervals. Nevertheless, it is worth remarking that multiple-target tracking could be supported in our architecture just by replacing the tracking algorithm with an alternative one, such as [13,15].

Each report is associated with a unique and sequential id by the sink. Notice that each group of data includes several sensor reports, being consecutive groups separated by a time interval (in seconds) defined by the user. Vector $\vec{P_e}$ contains the sequence of estimated intruder locations, being constantly updated based on the location associated with the different sensors ($\vec{P_s}$) that report the intruder's presence. To obtain the sequence of $\vec{P_e}$ values, the following operations are performed every time a new sensor report is received: Initially (first group created), the intruder location is estimated based on an exponential filter that weights new and old values. Factor α characterizes the behavior of this estimation: higher values make the system more responsive to abrupt changes in the position, whereas lower values make it more conservative.

From the second group created onward, all report groups are split into micro-groups, and both location ($\vec{P_{mgr}}$) and speed ($\vec{V_{mgr}}$)estimations are made for each microgroup. The estimated intruder location is again based on an exponential filter that weights microgroup location estimations ($\vec{P_{mgr}}$) and location estimations derived from speed ($\vec{P_{speed}}$). The latter is calculated as the projection of the previous estimated location ($\vec{P_e}[last\_id - 1]$) plus the distance provided by the velocity vector for the time elapsed between the two microgroups. Parameter α is dynamically calculated, increasing as the time difference between the current time and the initial microgroup time

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

increases. Parameter β (β∈[0, 1]) is used to regulate the growth of α, which must never grow beyond its upper bound (max_alpha ∈ [0, 1]).

Algorithm 1 Intruder tracking: position estimation process.

input: interval, max_alpha, beta
begin
set microint = interval/5; #microgroup interval
set last_id = 0;
set $\vec{P_e}[0] = \vec{P_s}[0]$; #initial location equal to the location of the first sensor reporting
for each report id received do {
if (timestamp[id] - timestamp[0] < $interval) { # for first group only
set $alpha = \frac{timestamp[id] - timestamp[id-1]}{timestamp[id] - timestamp[0]} \times beta$
if (alpha > max_alpha) set alpha = max_alpha;
set $\vec{P_e}[id] = \alpha \cdot \vec{P_s}[id-1] + (1-\alpha) \cdot \vec{P_s}[id]$
} else { # group-based estimation
if (timestamp[id]-timestamp[last_id+1] < microint) { # microgroup detected
set $\vec{P_{mgr}} = average(\vec{P_s}[last\_id+1] \text{ to } \vec{P_s}[id])$ #microgroup estimation
} else { # new microgroup
set $\vec{P_{mgr}} = \vec{P_s}[id]$; #est. group pos. equal to current sensor pos.
set last_id = id - 1;
}
set $\vec{V_{mgr}}$ = estimate_intruder_speed(last_id, id) #reports from last_id to id
set $\vec{P_{speed}} = \vec{P_e}[last\_id-1] + \vec{V_{mgr}} \times (timestamp[id] - timestamp[last\_id])$
set $alpha = \frac{timestamp[id] - timestamp[last-id]}{interval} \times beta$
if (alpha > max_alpha) set alpha = max_alpha;
set $\vec{P_e}[id] = \alpha \cdot \vec{P_{mgr}} + (1-\alpha) \cdot \vec{P_{speed}}$ }
} end

Our routing protocol is optimized to operate under the following assumptions, which are commonly met in our target scenario: (a) the number of sensor nodes does not increase over time; (b) sensor nodes remain at static positions; and (c) the sink is able to move freely throughout the sensed area, without constraints.The MRLG protocol distinguishes between three types of neighbor nodes, from the perspective of a particular sensor node: (1) downhill: includes those nodes that are closer to the sink (lower hop count); (2) peers: nodes at the same distance from the sink (similar hop count) and, (3) uphill: nodes further away from the sink (higher hop count). Notice that creating and maintaining the list of downhill, peer and uphill nodes becomes straightforward by listening to the hop count advertised in the messages broadcasted by these nodes, and comparing against the node's own hop count.

The sequence of actions taken by the MRLG routing algorithm when updating routes is the following:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

- ✓ Topology updates are triggered by the sink node. Since the sink can be mobile, it periodically sends Route Request (RREQ) messages to announce its presence at a rate that can be adjusted according to its degree of mobility (one second by default). These messages allow nearby sensor nodes to detect any changes in the sink's position, which may trigger a topology reconfiguration as explained below. The first message generated is fully propagated throughout the WSN, allowing the different sensor nodes to generate a vector field of routes pointing towards the sink node.
- ✓ Upon listening to the sink's RREQ messages, every sensor node stores the hop count value and the sequence number of the last sink message received; from the set of nodes that share the same (minimal) hop count towards the sink (downhill nodes), it picks one of them as the next-hop for data forwarding. Sensor nodes also store information about other neighbors (both peers and uphill nodes), based on information collected from all overheard messages. In summary, the information stored by each sensor is: <RREQ sequence number, hop count, next_hop, downhill nodes , peers , uphill nodes >.
- ✓ After processing RREQ messages, sensor nodes will conditionally rebroadcast them taking into account the hop count and the sequence number, according to Algorithm (2). In this algorithm, RREQ is an object that refers to the Route Request Packet, and Node is an object that refers to the data stored by the sensor itself; both objects contain attributes such as hop_count, which refers to the number of hops towards the sink, and seq_number, which is the sequence number generated by the sink (increases every time). Additionally, RREQ contains the ttl attribute, which refers to the Time-to-Live property, and source, which refers to the address of the sensor that broadcasted the packet. Node objects contain methods such as IsUphill(), to determine if a certain node is stored as an the uphill node, and UpdateNeighborList(), which allows updating the information about neighbor nodes by assigning them to the appropriate list: downhill, peers, or uphill.

VI. SENSOR SELECTION

We perform extensive simulations and compare our solution with to state of the art algorithms, using both real-world and synthetic data. In this module, we create the network and deploy it. Our network model consists of two entities with Client and Router. Where the module is developed using socket programming in C#.net. So we can execute the system in single system or multiple systems. In client system, we can target the location. In router node, we simulate Wireless Sensor Network nodes. We propose a range-free scheme called virtual-hop localization, which makes full use of local information to mitigate the non-uniform node distribution problem. Using virtual-hop, the initial estimated locations are more accurate than those output by other range-free schemes.

VII. SENSOR MOVEMENT

In this module, those identified good nodes are regarded as references and used to calibrate the location of bad ones. Links with different ranging quality are given different weights. Outliers in range measurements are tolerated using robust estimation.

VIII. TRACKING ERROR ESTIMATION

In order to evaluate the proposed algorithm and to assess its tracking accuracy, we devised a set of intruder mobility patterns (straight, random and curve), which are illustrated in Figure 2. We can observe in Figure 2, besides the intruder's path (represented as a slim dotted trajectory), some circles that highlight those sensors nodes that are activated at some instant of time, generating the appropriate reports for the sink. Crosses represent the sequence of estimations about the intruder's path made by the sink. Differences between real and estimated intruder trajectories are slightly more noticeable in Figure 2(b).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

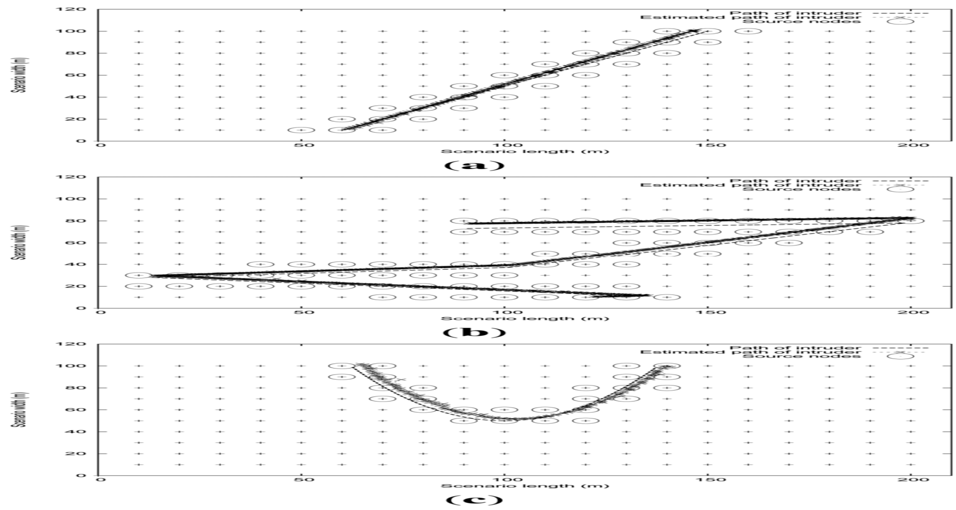

Figure 2. Illustration of the intruder tracking accuracy for different mobility patterns: (a) straight, (b) random and (c) curve.

Click here to enlarge figure To obtain an accurate measure of the overall error in determining the intruder's trajectory, the following actions are taken: Retrieve the exact intruder location at all times, based on the mobility pattern defined as input to the simulation.Based on the intruder reports received at the sink, filter all signals, keeping only those that correspond to the sensors that have detected an intruder. The available data are: the number of sensors that detected the intruder, the location of each sensor, and the intruder report time advertised by each sensor.

For each report received, obtain a new estimate for the intruder's location by relying on Algorithm (1), which creates information groups by combining the reports received at time intervals of fixed duration.
Calculate the mean value of the Euclidean distance between the sequence of known intruder coordinates and the intruder trajectory estimation during the entire test period.

In our evaluations, the values adopted for the different tracking algorithm parameters were: interval = 5 s, max_alfa = 0.25, and $\beta = 0.4$. Notice that, as shown in Figure 2, the filters and estimation techniques used in our proposed algorithm are able to achieve a noticeable performance in terms of trajectory estimation, even though simple binary sensors are used.
Concerning the calculated error, notice that three different factors are combined: (i) the intruder location inaccuracy associated with the binary reporting provided by the sensors, which will typically be equal to 10 meters for the first report generated, since this is the sensor's detection range; (ii) the mean estimation error introduced by the chosen data aggregation algorithm, based on the information received; and (iii) the delay experienced by the different reports, when traveling throughout the WSN, until the sink is reached.

IX. Conclusions and Future Work

The design of a WSN is influenced by many factors including hardware constraints, transmission media, energy consumption, topology, scalability and fault tolerance. The significance of these factors increases in environments with several hundreds or thousands of sensor nodes. Additionally, the protocols and algorithms adopted must be efficient and scalable as well.When targeting novel WSN applications, like real-time intruder tracking, information from different sources must be collected and processed as quickly as possible, to provide the sink with accurate information about the intruder location at all times. If, additionally, the sink needs to move along the WSN deployment scenario in an attempt to pursue and seize the intruder, the degree of complexity increases further, and the adopted routing protocol becomes critical.

In this paper, we introduced a comprehensive solution to achieve this goal when relying on cheap, binary detection sensors and the IEEE 802.15.4 technology. Our solution combines a fast intruder tracking algorithm with an efficient routing protocol (MRLG), to obtain the best performance. In particular, experimental results showed that, even at high

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

mobility levels for both the intruder and the sink (fast running speeds), the proposed strategy allows the tracking error to be typically maintained below 10 m, even for highly irregular intruder mobility patterns. Additionally, we compare the proposed MRLG routing protocol against CTP, a known tree-based protocol, to further emphasize on the benefits of our proposal. Overall, we consider that the proposed solution is validated by the results obtained.

REFERENCES

1 S. Martínez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, Apr. 2006.
2 F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration for tracking applications," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 61–72, Mar. 2002.
3 Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory Algorithms and Software*. Hoboken, NJ, USA: Wiley, 2001.
4 H. Wang, K. Yao, and D. Estrin, "Information-theoretic approaches for sensor selection and placement in sensor networks for target localization and tracking," *J. Commun. Netw.*, vol. 7, no. 4, pp. 438–449, Dec. 2005.
5 X. Cao, J. Chen, Y. Xiao, and Y. Sun, "Building environment control with wireless sensor and actuator networks: Centralized vs. distributed," *IEEE Trans. Ind. Electron.*, vol. 57, no. 11, pp. 3596–3605, Nov. 2010.
6 P. Cheng, J. Chen, F. Zhang, Y. Sun, and X. Shen, "A distributed TDMA scheduling algorithm for target tracking in ultrasonic sensor networks," *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 3836–3845, Sep. 2013.
7 R. C. Luo and T. M. Chen, "Autonomous mobile target tracking system based on grey-fuzzy control algorithm," *IEEE Trans. Ind. Electron.*, vol. 47, no. 4, pp. 920–931, Aug. 2000.
8 H. Song, V. Shin, and M. Jeon, "Mobile node localization using fusion prediction-based interacting multiple model in cricket sensor network," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4349–4359, Nov. 2012.