

# Dead Reckoning Based Target Tracking in Wireless Sensor Networks

Muhammad Taqi Raza<sup>†</sup>, Gargi Bag<sup>‡</sup>, Seung-Wha Yoo<sup>‡</sup>, Ki-Hyung Kim<sup>‡</sup>

<sup>†</sup>USN Networking Research Team

<sup>†</sup>Electronics and Telecommunications Research Institute (ETRI), South Korea

<sup>‡</sup>Information and Communication Department

<sup>‡</sup>Ajou University, South Korea

<sup>†</sup>taqi@etri.re.kr

<sup>‡</sup>{gargibag, swyoo, kkim86}@ajou.ac.kr

**Abstract**— *Target tracking is one of the key applications of Wireless Sensor Networks (WSNs) that forms basis for numerous other applications. Real situations in tracking, such as an unexpected change in the mobile event's direction, failure of event detection, or transmission failure of an error message may result into erroneous tracking information. Thus, in order to prevent and handle errors effectively, it is required that tracking algorithm must incorporate error avoidance and error correction phase. In this paper we present Dead Reckoning (DR) based target tracking protocol. DR determines target's present position by projecting its past positions and speed over elapsed time and known target's path. Moreover, it inherits the functionality of error correction and error avoidance using a position fix technique. The performance of DR based tracking is also discussed, which confirms the efficacy of the algorithm.*

**Keywords:** *Sensor network, sensor nodes, target tracking, Dead Reckoning (DR), Position Fix*

## 1. INTRODUCTION

A sensor node consists of a sensor board, a processor with limited computational power, memory, transmission circuitry, and a small fixed size battery source, [1][2][3]. Sensor nodes are distributed over a Region of Interest (RoI), and connected to the base station via a command network, hence forming a sensor network [4]. These sensor networks are being used for a variety of purposes including both military and commercial applications.

Target tracking in a sensor field has become one of the most important applications of such sensor networks particularly for military purposes. Here, the main objective is to utilize sensor nodes *within the proximity* of the target to localize and track it. Also, it is required to provide a complete picture of tracking information to the base station, so that appropriate measures could be taken. The tracking process may be affected by the hostile network conditions in the war field. Thus the tracking algorithm must be able to provide self diagnosis of tracking errors and should be able to adjust accordingly. Many solutions have been proposed for target tracking, with an avid focus on the battery conservation of

sensor nodes. These solutions require identification of the target's movement patterns, its history records, and temporal mobility behavior. Such approaches are highly computational, and intense on storage. Moreover these approaches lack in providing a mechanism to handle tracking errors, and the process to delegate tracking information to the base station. In this paper, we propose Dead Reckoning (DR) based target tracking in WSN. Our approach is inspired by the well known navigation protocol, i.e. DR, described as *the process of estimating object's current position based upon a previously determined position, or fix, and advancing that position based upon known speed, elapsed time, and course* [5].

We assume triangulation based localization technique, where three sensors take part to localize the target. Then the location information is sent to the cluster head. The cluster head runs DR based tracking algorithm and, if required, it sends tracking information to the base station. Based on this information, the base station plots the tracking path in order to get the overall picture of the target course, so that the appropriate measures could be adopted accordingly. The error correction and error avoidance phase starts when the cluster head receives the *fix*, also known as *position fix*. *Fix* is the known position within the sensor field, against which target's current position is measured. Using position fix the cluster head observes an accurate target location and verifies it too. At the other hand, the base station adjusts the location errors on every position fix. Significant tracking accuracy is achieved with error correction and error avoidance phase of DR.

The rest of the paper is organized as follows. Section 2 describes the related work. The system model and assumptions are discussed in section 3. We present DR based target tracking process in section 4, followed by the performance evaluation of the proposed scheme in section 5. Finally, Section 6 concludes the paper.

## 2. RELATED WORK

We have divided our related work into two parts. First part discusses various approaches that have been used for target tracking, whereas the second part gives an overview of DR applications in WSN.

Significant work has been done in the area of target tracking. Some approaches for target prediction and tracking are as follow:

In [6], we proposed target tracking process through triangulation method. This approach addressed localization based tracking. In [7], we introduced target prediction mechanism through yaw rate. The proposed approach takes into consideration the physical constraints on randomness of mobility. The scheme, however, introduces relatively higher prediction errors in case of variable target speeds and over random paths.

In tree based collaboration [8][9] methods, sensor nodes are deployed in such a manner that the tree is dynamically formed by adding some nodes and pruning other nodes as the target moves. By aggregating the data at a node closer to the target, the network traffic can be reduced and the energy consumption can be optimized. But in this scheme, a node can only send data to its parent. As the target moves further in the sensor network, sending data to the parent may cause a lot of traffic over the network that results in significant energy drain. This problem can be addressed through the tree reconfiguration, that is, by utilizing an optimized complete reconfiguration scheme and an optimized interception-based reconfiguration scheme. Reconfiguring the trees always put extra computational and transmission overhead on the network. Furthermore, as the network density grows, reconfiguration becomes cumbersome. These issues minimize the network life time considerably.

Other methods such as [10][11] organize the sensors into clusters and use a normal beam or a high beam to declare the presence and absence of a target. When the cluster is active the normal beam is used, whereas the high beam is only used when the target is lost. The messages are exchanged between cluster heads. Each cluster head activates the next appropriate cluster before the target arrives. The main disadvantage of cluster-based target tracking is the overhead for selecting a cluster head, because the cluster head acts as a central authority for other nodes and can also leads to a single point of failure. This could be addressed by rotating the role of cluster head [12], i.e., every node should take the role of being a cluster head during periodic cycles.

Other strategies like activation, randomized activation, and selective activation, as described in [13], are all focusing on trajectory prediction. These describe that the energy consumption can be reduced when appropriate selective activation is used with a yielding prediction. These approaches involve a large number of sensors with frequent exchange of data with the base station. We assert, however that a tracking mechanism for target location is needed that utilizes lesser resources, lesser degree of communication and provides a higher degree of accuracy.

At the other hand, many applications in WSN are using position fix and are employing DR based systems.

As in [14], source location estimation using passive and stationary acoustic sensors (microphones) is obtained. The ground truth is obtained by interpolating an on-board GPS recordings that records a position fix every 15 seconds.

In [15] the location of a sensor (node or device) is obtained from the relevant information obtained from reference nodes, i.e. beacons. The location process is repeated once the fix is obtained. For more accuracy four-dimensional fix is used.

In [16] Dead Reckoning Systems (DRS) is adopted to tackle satellite visibility problems in GPS positioning. A low-cost DRS containing a wheel speed sensor, a digital compass or gyroscope to provide distance and heading information for position prediction in case insufficient satellites are available for position determination.

As the matter of fact, none of the above approaches can be used for target tracking. Because the target in the war field is anonymous and does not employ GPS chip. Moreover these approaches are computational hungry too.

### 3. SYSTEM MODEL AND ASSUMPTIONS

In this section, we formulate and enlist fundamental assumptions needed for ensuring the plausibility of our target tracking protocol.

- *Tracking model:* The sensor nodes can track one and only one mobile target at one time. This restriction makes the scope of the paper fundamentally different from multiple target detection and tracking problem.
- *Mobility model:* A mobile target exhibits Smooth Random Mobility Model, where both the speed and movement direction of the target is partly decided by their previous values [17]. The step size of movement is less than the smallest hop inside the network.
- Each sensor is equipped with a radio-triggered hardware component that activates sensors from/to the sleep and quasi-sleep states, by sending a special wake up radio signal [18].
- Each sensor in a sensor field has the same transmission signal strength and receiver sensitivity.
- Three sensors localize the target using triangulation method.
- The nodes are densely deployed to an extent that their transmission and sensing ranges overlap.

### 4. DEAD RECKONING-BASED TARGET TRACKING

In this section, we start with a state-based description of our DR based target tracking mechanism that activates a minimum number of nodes to track a mobile object. We believe, in accordance with the study done in [19] that the operational lifetime of the network may be improved manifold by using DR based target tracking.

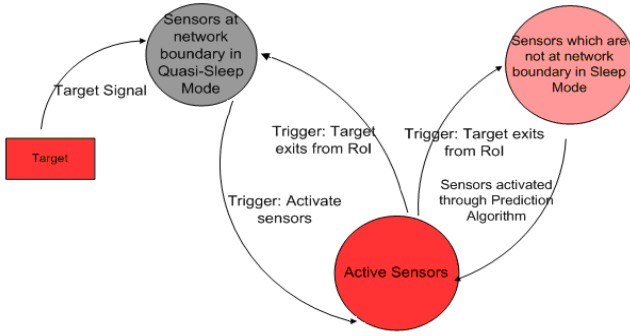


Fig. 1: State transition diagram of sensor states

Figure 1 shows the state transition diagram of an arbitrary sensor node when the object is absent, when it enters the sensor network RoI, and while being tracked. In the absence of an object, all the sensors along the boundary of RoI are in quasi-sleep state, while nodes deeper inside the network are in sleep state. On detecting a target, sensor nodes resume an active state to triangulate the location of the object, in Cartesian Coordinates. A trigger is used to activate the closest nodes only, along the tracking trajectory, in order to let the maximum number of nodes sleep.

#### 4.1. TARGET TRACKING

*Dead reckoning allows for determining target's present position by projecting its past courses steered and speeds from a known past position* [5]. The sensor nodes localize the target and send the measured location coordinates to the cluster head. The cluster head processes the information, time stamp it<sup>1</sup>, and then sends this information to the base station, when either of this is true.

1. At every time stamp ( $TS_n$ , where  $n=1\dots m$ ).
2. After every change of course or speed.
3. After every fix or running fix.

Figure 2 and 3 best describes the dead reckoning based target tracking procedure. Figure 2 shows that when the target is within the sensor field, then the boundary sensor nodes are in sleep state. Three sensor nodes, which are closer to the target, triangulate the target and measure<sup>2</sup> the instantaneous speed of the target as given in Equation (1). Then the closest sensor towards the cluster head sends the measured location coordinates and the measured speed of the target to the cluster head. Then the cluster head determines whether any of the above mentioned rules is true or not.

<sup>1</sup> When either of rules 2 or 3 occurs, the cluster head time stamps the event (though timer is not expired yet). After this time stamp, the time stamp counter restarts.

<sup>2</sup> Localization is made periodically. This period  $\ll$  Time stamp (TS)

$$Speed = \frac{\text{Current Target Location}(x; y) - \text{Previous Target Location}(x; y)}{\text{Current Time} - \text{Prev Time} ('t')} \quad (1)$$

As shown in Figure 3, the target tracking process is initiated at time stamp  $TS_1$ . The position determined at  $TS_1$  is called the **initiated position**. At  $TS_1$ , the cluster head sends target's initial tracking information<sup>3</sup> to the base station. The target moves on with the same speed and on the same course. At  $TS_2$ , cluster head observe the rule 1, requiring the tracking information to be sent to the base station. Then it sends the latest tracking information to the base station. When target moves further and changes its direction<sup>4</sup> (i.e. towards South-East, as shown in Figure 3) then the cluster head observes rule 2. The cluster head time stamps as  $TS_3$  and sends tracking information to the base station. After some time, cluster head observes a change in DR position and speed, i.e. a course of  $080^\circ$  and a speed of 20 m/sec. It time stamps as  $TS_4$ , and sends target observed information to the base station. At  $TS_5$ , there is not only change in course but also change in DR position. The cluster head observe a course of  $090^\circ$  and a speed of 10 m/sec and sends this information to the base station. Before the expiration of time stamp timer (i.e.  $TS_6$ ), the cluster head observes change of course and speed, i.e. a course of  $140^\circ$  and a speed of 30 m/sec. It then time stamps as  $TS_6$ , and sends tracking information to the base station. The target moves on and then not only  $TS_7$  occurs but also target reduces its speed to 20 m/sec. Hence both rules 1 and 2 are satisfied, that require cluster head to send the tracking information to the base station. Note that at  $TS_7$ , second position fix is observed (i.e. rule 3) which leads to the error correction and error avoidance phase (as explained in Section 4.2). Further more, the cluster head observes the change in course (i.e.  $110^\circ$ ). It time stamps as  $TS_8$  and sends the tracking information to the base station. At the other end, the base station receives the tracking information from the cluster head and plots the DR positions accordingly. Plotting helps the base station to get the whole picture of the track and the behavior of the target.

<sup>3</sup> Tracking information includes DR position, course information and target speed

<sup>4</sup> The cluster head can determine the change in direction by observing a couple of previous target locations.

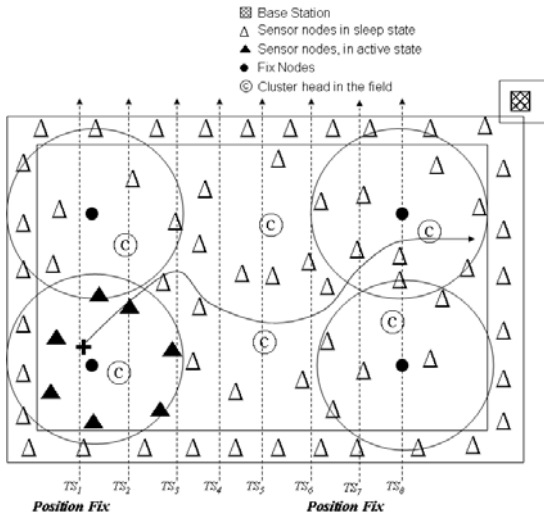


Fig. 2: Dead Reckoning based target tracking procedure

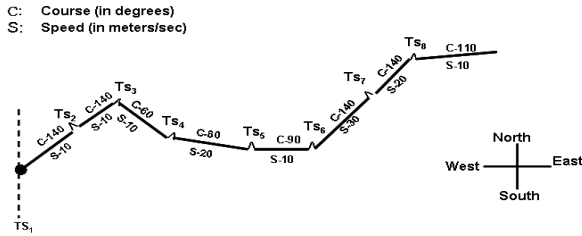


Fig. 3: Dead Reckoning plotting by base station

## 4.2. ERROR CORRECTION AND ERROR AVOIDANCE USING POSITION FIX

Unpredictable movement of target causes event to occur randomly and occasionally. Continuous activation of sensor nodes wastes a significant amount of energy. Thus energy-efficient tracking technique must be adopted by turning off the radio transmission or lengthening the system sleep-time in the absence of an event. Hence we emphasize that next three sensors, to be activated, must lie within the possible drift<sup>5</sup> of the target. The key issue is that the target can deviate from the possible drift range, which results in location error/miss. Hence the location error/miss unavoidably occurs due to unexpected directional and speed changes of a moving event at random time. When the error occurs, the tracking process is no longer functional because the sensor nodes located near the moving event were likely to turn off their sensors and radios to conserve energy. In the worst case, the failure of sensor nodes can occur at any time due to power depletion or harsh natural environments; hence, a moving event is able to slip away from the wake-up zone without detection. In this case, the effort both to conserve energy and to obtain acceptable performance leads to failure. Unfortunately, most of the target tracking

<sup>5</sup> We assume that target can deviate within the range of  $+90^\circ$  and  $-90^\circ$

schemes overlook this and pay a little attention to dealing with location error/miss in target tracking process.

DR scheme provides a mechanism for the error correction and error avoidance, known as **position fix or resetting DR** [5]. Fix is the known position within the sensor field, against which target's current position is measured. At the time of deployment a number of fix nodes are also deployed within the region (as shown in Figure 2). In order to conserve energy, these fix nodes periodically wake up in order to determine the presence of the target and then go to sleep again until the next frame. Here we have assumed that the duty cycling of fix nodes are small enough to detect the target at its certain range of speed. When any of the fix nodes determine the presence of the target, then it needs to localize the target. As a fix node cannot localize the target thus it sends wakeup radio triggered broadcast to all the sensor nodes within its vicinity. These sensor nodes (in pair of three sensors) localize the target on the behalf of the fix node and send their readings to the cluster head. The cluster head receives the location information from multiple sensor nodes and matches it for verification. Once the location information is verified, then it sends this DR position to the base station. Sending DR position of the target to the base station is extraneous to the scope of this paper. Many approaches have already been proposed that address data communication between the cluster head and the base station as in [20][21].

Moreover the cluster head sends the radio triggered multicast to all of its associated sensor nodes (except those three sensor nodes which are closer to the target and part of the normal DR based tracking procedure) in order to let them sleep. Note that first position fix is determined when the target tracking process starts (as shown in Figure 2, at TS<sub>1</sub>).

Here it can be argued that the fixed nodes' responsibility could be delegated to the cluster head. It should be noted that the number of cluster heads to be deployed are related to the network topology and the number of sensor nodes. Because the main role of cluster head is to enable the communication between the sensor nodes and the base station; whereas the deployment of fixed nodes is related to how often error correction phase is required. If there are more number of fixed nodes then more often error correction phase would be repeated. The error correction and error avoidance effect with respect to the number of fixed nodes have been discussed in the performance evaluation section, i.e. Section 5.

The base station must take precautions to ensure that all hazards to track the target along its path are accounted by the approximate nature of a DR position. One method which can be used is **fix expansion** [5].

Fix expansion takes into account possible errors in the DR calculation caused by factors which tend to affect the target's actual location/course and speed. The base station considers all such factors and develops an expanding "error circle" around the DR plot. One of the basic assumptions of fix expansion is that the various individual effects of

location, speed, and noise error combine to cause a cumulative error which increases over time – hence, the concept of *expansion*. Errors considered in the calculation of the fix expansion encompass all of the errors that can lead to DR inaccuracy. Any method which attempts to determine an error circle must take these factors into account. The base station can use the magnitude of set and drift calculated from its DR plot. The base station determines the effect, that each of these errors has on its course and speed, and applies that error to the fix expansion calculation. As noted above, the error is a function of time; it grows as the target proceeds down the track without obtaining a fix. Therefore, the base station must incorporate its calculated errors into an **error circle** whose radius grows with time. For example, assume the base station calculates that all the various sources of error can create a cumulative position error of no more than 2 cm. Then its fix expansion error circle would grow at that rate; it would be 2 cm after the  $TS_1$ , 4 cm after the  $TS_2$ , and so on. At what value should the base station start this error circle? Note that a DR is laid out from every fix. All fix sources have a finite absolute accuracy, and the initial error circle should reflect that accuracy. *Assume, for example*, that the target tracking process has an *accuracy of 0.5 cm*. Then the initial error circle around that fix should be set at 0.5 cm. The error circle is constructed as follows. **When the base station obtains a fix, reset the DR to that fix.** Then, enclose that DR position in a circle, the radius of which is equal to the accuracy of the procedure used to obtain the fix. Then it lays out the ordered course and speed from the fix position. Then base station applies the fix expansion circle to the  $TS_n$  of DR. In the example given above, the DR after  $TS_1$  would be enclosed by a circle of radius 2.5 cm, after  $TS_2$  4.5 cm, and so on. Having encircled the four time stamp ( $TS_4$ ) DR positions with the error circles, the base station then observes two lines originating tangent to the original error circle and simultaneously tangent to the other error circles. The base station then examines the area between the two tangent lines for hazards to tracking. This technique is illustrated in Figure 4 below. The fix expansion encompasses *all the area in which the target could be located* (as long as all sources of error are considered). The fix expansion may grow at such a rate that it becomes unwieldy. Obviously, if the fix expansion grows to cover too large an area, it has lost its usefulness as a tool for the tracking, and he should obtain a new fix. Hence in this case, the **error avoidance** is required for tracking accuracy. For error avoidance, the base station keeps the average of fix expansion, for each time stamp. Then it adds this average error into the observed reading in order to keep the tracking path close to the actual one. Note that by incorporating the average error into the observed reading, we can decrease the error but cannot remove it.

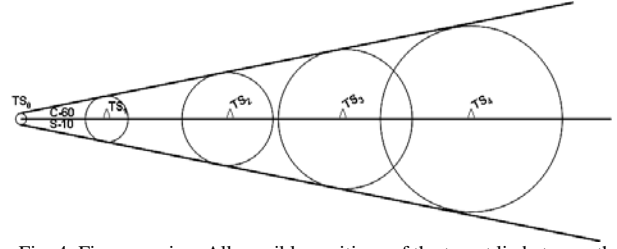


Fig. 4: Fix expansion. All possible positions of the target lie between the lines tangent to the expanding circles. The area is adjusted for error correction.

## 5. PERFORMANCE EVALUATION

We evaluate the proposed DR based tracking mechanism by developing a complete simulation in MATLAB. The surveillance area is  $192 \times 192 m^2$ . A total of 144 nodes are deployed in a  $12 \times 12$  logical grid. The main reason of dividing the whole area into a grid is to examine the target behavior at each step. The average speed of the target is 19.6 m/sec. The performance metrics of interest are (a) the tracking error: the deviation of the measured location from the exact location of the target; (b) target lost event: the behavior of our protocol when the target is lost (Section 5.1), and (c) the effect of position fix. Each data point reported below is an average of 200 simulation runs.

We benchmark the trace route and performance metrics as explained in [22]. Figure 5, shows the performance comparison between the proposed mechanism and the basic mechanism<sup>6</sup> [22], which provides only a simple tracking technique without error handling. The traces show that the tracking data generated with the proposed mechanism are more accurate than those of the existing mechanism.

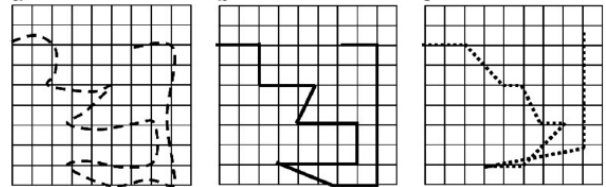


Fig.5: (a) Actual trace (b) our mechanism (c) Basic mechanism]

Figure 6 shows the tracking errors against the number of turns that the target takes, for the proposed approach and the basic scheme. The tracking error is represented as an error ratio, which is defined as the number of sensor nodes that missed the event, over the number of nodes that were supposed to detect the event. As in case of the proposed approach, target random turns don't affect the efficacy of tracking. Whereas for the basic mechanism a substantial

<sup>6</sup> The term “basic mechanism/scheme”, as described in [22], is used for a scheme in which message to wake-up sensors is propagated within a certain range, centered on the expected target location; thus, a slight change in the target's direction does not cause an error.

degree of errors, regardless of the size of the wake-up zone, are observed.

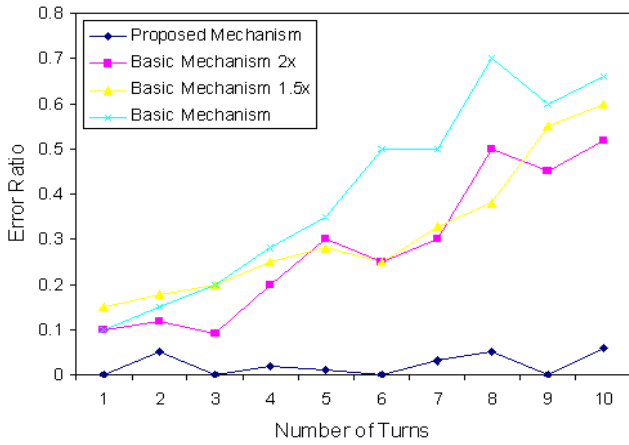


Fig. 6: Direction changes vs. delivery error ratio

Through triangulation method, three sensors will always be active for target localization. But there can be the case when the location miss occurs due to the tracking error (because for a long time no position fix occurs). In order to address such a scenario, we assume that the sensor may sleep for 1 sec, 1.5 sec, 2 sec or more. Figure 7 shows the average values of successful receiving rates. By using our approach, almost 80% accuracy is guaranteed with 2 sec more sleep time. But with basic mechanism, 42%, 29% and 18 % accuracy rates are observed against 1.0 sec, 1.5 sec and 2 sec more sleep-time periods, respectively.

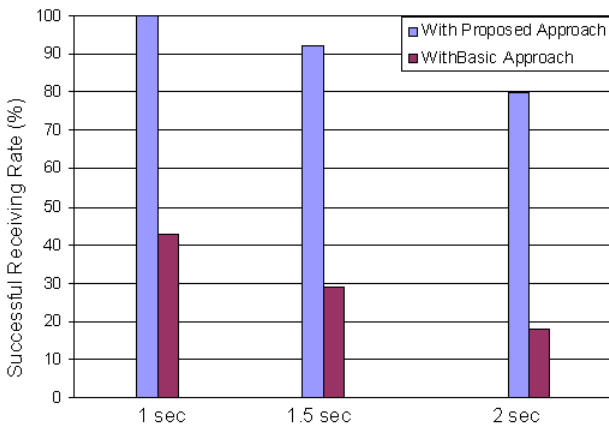


Fig. 7: Successful receiving rate with 1.0 sec. and 1.5 sec sleep time

The error correction and error avoidance phase is directly proportional to the number of time position fix is observed. By installing more number of position fix nodes, we can enhance the probability of position fix occurrence. Moreover the target moving with relatively high speed will also lead to the more occurrences of position fix. As shown in Figure 8, we have observed the time taken to observe the position fix by varying the speed of the target and the number of position fix nodes. It can be seen that when the

target moves fast and the number of fix nodes are more, then the position fix is observed more often. As we can't control the target's speed in the war field, but we can reduce the error by installing relatively more number of position fix nodes. Now the question arises that what is that number? The number of position fix nodes depends on the minimum time interval after which position fix is required; and what could be the minimum possible speed of the target. As shown in Figure 8, 04 fix nodes are required in order to get a position fix after every 30 sec, when the target is moving with at least 10 m/sec or more.

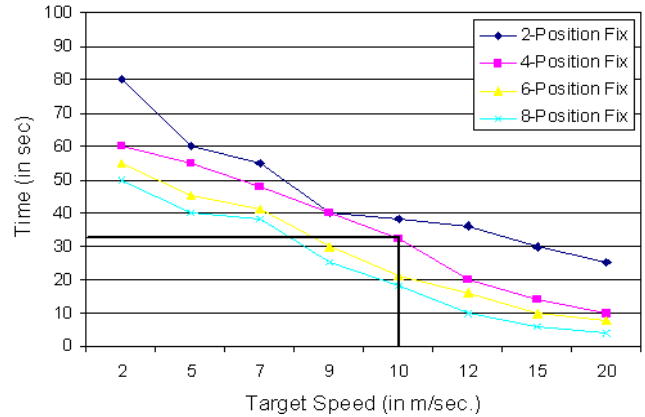


Fig. 8: Time to obtain the Fix with variable target speed

### 5.1. SUDDEN STOP

We may also face uncertain and unnatural behaviors of the target, in which the target reduces its speed suddenly, and then either it stops moving further or it gets a turn of 180°. Under these conditions, our algorithm might not work properly, where the target decrease of speed is instantaneously. To address this problem, the concept of timer is introduced, where the activated sensors set a timer. The sensors wait for the target until the timer is expired. When the timer is expired, these sensors nodes report this "target lost" error back to the cluster head. The cluster head activate all of its associated sensor nodes. Hence all the sensor nodes are activated near the location, where the target is lost. Once the lost target is found, then the target is localized and all other activated sensors will switch to the sleep state, except for those nodes which localize the target. At this point, our mechanism shall resume its normal behavior of tracking.

## 6. CONCLUSION

In this paper, a mechanism is devised, which is based on the phenomenon of dead reckoning. We build our network model, by elaborating all the steps involved in target tracking. Two different sleep states i.e. sleep and quasi-sleep states are introduced to conserve the overall energy of the network in the absence of the target. The paper discusses in breadth the DR based target tracking process. Error correction and error avoidance is part of DR based tracking,

that is done by using position fix. We discuss performance of our proposed scheme in terms of tracking error, node sleep time and target lost.

### ACKNOWLEDGEMENT

This work is supported by the R&D program under Ministry of Knowledge Economy (MKE) of the Republic of Korea. We are also thankful to MIC/IITA for their support to authors towards the development of IP based sensor network (IP-USN) technology.

### REFERENCES

- [1] A. Sinha, A. Chandrakasan. "Dynamic Power Management in Wireless Sensor networks", IEEE Design and Test of Computers, 2001
- [2] Yongjun Xu, Lingyi Liu, Peifu Shen, Tao Lv, Xiaowei Li, "Low power processor design for wireless sensor network applications", IEEE Wireless Communications, Networking and Mobile Computing, 2005
- [3] Zhi-Yan Cao, Zheng-Zhou Ji, Ming-Zeng Hu, "An image sensor node for wireless sensor networks", IEEE International Information Technology Conference, 2005
- [4] Richard R. Brooks, Parameswaran Ramanathan, and Akbar M. Sayeed, "Distributed Target Classification and Tracking in Sensor Networks", Proceedings of the IEEE, 2003
- [5] Bowditch "The American Practical Navigator", National Imagery and Mapping Agency, Bethesda – Maryland, 1995, page 113-118
- [6] Muhammad Taqi Raza H. M., N. D. Gohar, et al., "Distributive Target Localization and Tracking in Sensor networks", High Capacity Optical Networks and Enabling Technologies (HONET), UNCC USA, 2006
- [7] Muhammad Taqi Raza H. M., Ali Hammad Akbar, et al., "A Yaw Rate Aware Sensor Wakeup Protocol (YAP) For Target Prediction And Tracking In Sensor Networks", MilCom, Oct. 2007, Orlando (Florida), USA
- [8] H. T. Kung and D. Vlah, "Efficient Location Tracking Using Sensor Networks," Proc., IEEE WCNC, 2003
- [9] W. Zhang and G. Cao, "Optimizing Tree Reconfiguration for Mobile Target Tracking in Sensor Networks," IEEE Transactions on Wireless Communication, 2004
- [10] Francine Lalooses, Hengky Susanto, and Chornng Hwa Chang, "Recovery Target Tracking in Wildlife", Technical Report, Electrical Engineering Department, Tufts University, 2005
- [11] H. Yang and B. Sikdar. "A Protocol for Tracking Mobile Targets using Sensor Networks", 1<sup>st</sup> IEEE Workshop on Sensor Network Protocols and Applications, 2003
- [12] S. Bandyopadhyay and E. Coyle. "Minimizing Communication Cost in Hierarchically Clustered Network of Wireless Sensors", IEEE WCNC 2003
- [13] Sundeep Patten, Sameera Poduri, and Bhaskar Krishnamachari, "Energy-Quality Tradeoffs for Target Tracking in Wireless Sensor Networks", International Symposium on Aerospace/Defense sensing Simulation and Controls, Aerosense (IPSN), 2003
- [14] Xiaohong Sheng and Yu-Hen Hu, "Maximum Likelihood Wireless Sensor Network Source Localization Using Acoustic Signal Energy Measurements", IEEE Transactions on Signal Processing, 2003
- [15] Peter Brida, Jan Duha and Marek Krasnovsky "On the Accuracy of Weighted Proximity Based Localization in Wireless Sensor Networks", Personal Wireless Communication, 2007
- [16] Mok Chi-ming Esmond, "Ubiquitous Positioning Technologies and LBS", Location Magazine, 2007
- [17] C. Bettstetter, "Smooth is Better than Sharp: A Random Mobility Model for Simulation of Wireless Networks", ACM Intern. Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM), Rome, Italy, July 2001
- [18] Lin Gu and John A. Stankovic, "Radio-Triggered Wake-Up for Wireless Sensor Networks", 10th IEEE Real-Time and Embedded Technology and Applications Symposium, 2004.
- [19] Jason L. Hill, David E. Culler, "MICA: A Wireless Platform for Deeply Embedded Networks", a Technical Report on Power consumption of mice hardware, UC Berklay, 2002
- [20] Dawei Xia, et al. "Near-Optimal Node Clustering in Wireless Sensor Networks for Environment Monitoring", IEEE CCECE/CCGEI, 2006
- [21] J F Shi et al., "Study on Communication Mode of Wireless Sensor Networks Based on Effective Result", Journal of Physics: Conference Series 48 (2006) 1317–1321
- [22] Sung-Min Lee et al., "Energy-aware location error handling for object tracking applications in wireless sensor networks ", Elsevier Computer Communications, 2007