A YAW RATE AWARE SENSOR WAKEUP PROTOCOL (YAP) FOR TARGET PREDICTION AND TRACKING IN SENSOR NETWORKS

Muhammad Taqi Raza H. M., Ali Hammad Akbar, Shafique Ahmad Chaudhry, Gargi Bag, Seung-wha Yoo, Ki-Hyung Kim
Department of Information and Communication Engineering
Ajou University, Republic of Korea
{taqi, hammad, shafique, gargi, swyoo, kkim86}@ajou.ac.kr

Abstract— *Target tracking is one of the key military applications of wireless sensor networks (WSN). The overall procedure involves target detection, localization, and target tracking. Because of the WSNs' resource (especially energy) constraints, it is desired that target tracking could be done by involving as less number of sensor nodes as possible. Due to the uncertain behavior of the target, this goal is hard to achieve without predicting future target location, so that only selected sensors are activated before the target reaches a future location. This prior activation contributes to increase the overall lifetime of sensor networks. In this paper, a yaw aware sensor wakeup protocol (YAP) is proposed for prediction of future target location. The algorithm is distributive in nature and selects appropriate sensors to determine the target track, such that target track is obtained even if the predicted location differs from the actual one. The performance of YAP is also discussed on different target tracks, which confirms the efficacy of the algorithm.*

Key words: Sensor network, sensor nodes, target prediction, target tracking

1. INTRODUCTION

A sensor node consists of a processor with some limited computational power and memory, low battery power, and transmission circuitry [1-3]. 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].

Target tracking in a sensor field has become one of the most important applications in ubiquitous sensor networks for military purposes. The main objective of the tracking is to activate appropriate sensors before the target reaches their neighborhood. As the target can move to any path depending upon its speed, the real challenge is that which sensors will take part for tracking, by keeping in view sensor nodes' constraints like low power, low memory and less computational ability. Many solutions have been proposed, for prediction of target future location, which requires identification of target movement patterns, its history records etc. These all approaches are highly computational, and storage hungry. In order to predict future location on the basis of target history, the sensor nodes need to send data to the base station, periodically, (where the base station could maintain all these parameters and could predict the future target location). This process creates another issue of the data aggregation which increases nodes' awake time, consequently reducing network life time. So keeping in view the issues about future prediction, we propose an algorithm, where there is neither a need to maintain the target history records nor to maintain its path patterns. We build our idea on the phenomena of yaw rate, which can be described as the rate at which the object deviates from its path and conventionally known as the object's rotation on its central axis. In our approach, we predict future target location by considering the speed of the target, and its drift from the original path, and incorporate its current behavior for next prediction. Our prediction is based on the concept that every curve is the combination of a number of straight lines with varying length, and due to the target's freedom of movement, if we consider the target speed and drift then an accurate prediction can be made. Our results show that YAP shows better performance under different target paths.

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 YAP in section 4, followed by the performance evaluation in section 5. Section 6 concludes the paper.

2. RELATED WORK

Significant work has been done in the area of target tracking. In this section, we shall discuss various approaches that have been used.

In tree based collaboration [5-6] method, sensor nodes are deployed in such a manner that the tree is

1-4244-1513-06/07/$25.00 ©2007 IEEE

dynamically formed by adding some nodes and pruning other nodes as the target moves. By aggregating the data at a node close to the target, the network traffic can be reduced and the energy consumption can be optimized. The problem is that a node can only send data to its parent, and if target moves further in the sensor network, the root of the tree for sensor nodes can be far away depending upon the target's location, and hence sending data to the root may cause lots of traffic over the network, which costs a large amount of energy. This problem can be addressed through the tree reconfiguration, through an optimized complete reconfiguration scheme and an optimized interception-based reconfiguration scheme. But reconfiguring the trees always put extra computational and transmission overhead on the network. Also as the network density grows, the reconfiguration becomes cumbersome. These issues minimizes the network life time.

Another method is to organize the sensors into clusters and use normal beam or a high beam, as described in [7-8]. When the cluster is active the normal beam is used, whereas high beam is only used when the target is lost. The messages are exchanged between cluster heads. Each cluster head activates the appropriate next cluster before the target arrives. The main disadvantage of cluster based target tracking is overhead for becoming a cluster head, because the cluster head acts as a central authority for other nodes and also lead to single point of failure. But this could be addressed by rotating the role of cluster head [10], i.e., every node should take turns being a cluster head during every periodic cycle. As the sensor nodes delegate their responsibility to the cluster head with which they are associated, this can lead to the single point of failure. Moreover this approach may require more variant to predict future target path and target location, which could bring more complexity that yields computational overhead.

Other strategies like activation, randomized activation, and selective activation, as described in [9] are based on trajectory prediction. This work describes that if selective activation is used with a good prediction algorithm then energy consumption can be optimized. This approach involves a large number of sensors for the exchange of data to the base station. We assert that a prediction algorithm which is less resource hungry still needed in the sensor networks.

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 depicts random way point mobility. The step size of movement is less than the smallest hop inside the network. A compromise on this assumption takes a direct toll on the accuracy of the tracking operation.
- Each sensor node knows its location on the two dimensional sensor network Region of Interest (RoI) in Cartesian Coordinates.
- Each sensor in a sensor field has the same transmission signal strength and receiver sensitivity.
- The transmission and sensing ranges are the same. This assumption is necessary to ensure there are no holes in RoI [11].
- The nodes are densely deployed to an extent that their transmission and sensing ranges overlap.
- The node closer to the target activates next sensors, based on the predicted location.
- The sensor nodes exhibit three states of activation as shown in Table 1.

Table 1: Active and idle power consumption for Mica hardware

Sensor node module/ state	Symbol	Microcontroller, and memory	Sensor	Transceiver
Active	☺	✓	✓	✓
Quasi sleep	😐	✗	✓	✗
Sleep state	☹	✗	✗	✗

4. EFFICIENT PREDICTION BASED TARGET TRACKING

In this section, we start with a holistic description of our proposal of Yaw Rate Aware Sensor Wakeup Protocol (YAP) that awakes a minimum number of nodes to track a mobile object. YAW can be described as the rate at which the object deviates from its path and conventionally known as the object's rotation on its central axis. We believe, in accordance with the study done in [13] that the operational lifetime of the network may be improved manifold by using YAP. Later in the sub-sections, we discuss the operation of YAP during the target absence, detection and tracking respectively.

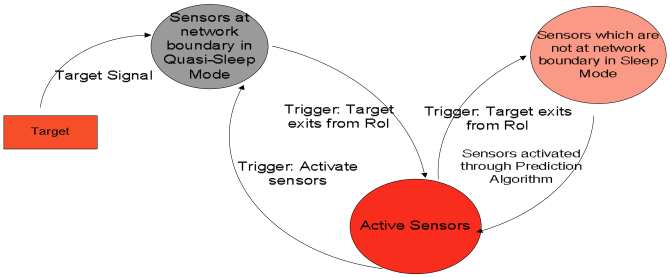

Figure 1: State transition diagram of sensor states

Figure 1 shows the state transition diagram of an arbitrary sensor node as 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 along the predicted trajectory in order to let the maximum number of nodes sleep.

4.1. TARGET ABSENCE

When there is no target in the sensor network RoI, the sensor nodes at the boundary of the sensor network are in quasi-sleep state, while all the nodes deeper in the network are in sleep mode (as shown in Figure 2).

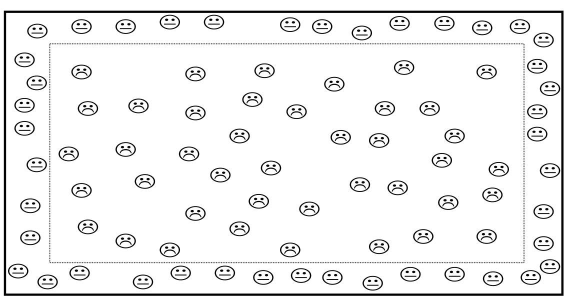

Figure 2: Initial state of sensor nodes in the absence of a target

4.2. TARGET DETECTION

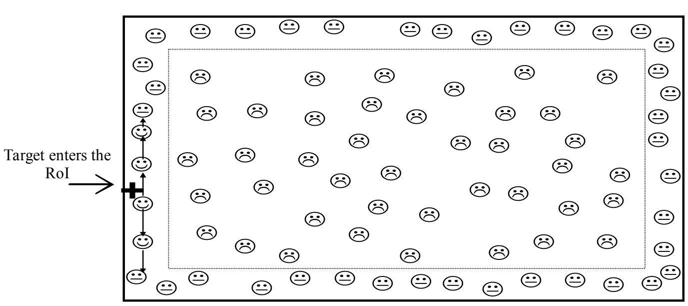

Figure 3: A boundary node detects a target and disseminates a directed trigger broadcast

When a mobile target enters the RoI, it is detected by sensor nodes which triangulate it. These nodes predict the next location of the target using YAP, and disseminate a directed trigger broadcast to notify all the nodes in the quasi-sleep mode, in order to let them sleep.

4.3. TARGET LOCALIZATION

$(X - x_1)^2 + (Y - y_1)^2 = d_1^2$ (a)
$(X - x_2)^2 + (Y - y_2)^2 = d_2^2$ (b)
$(X - x_3)^2 + (Y - y_3)^2 = d_3^2$ (c)

Solving equations (1), (2) and (3) for X and Y:

$$X = \frac{(-b2c1 + b1c2)}{(b2a1 - b1a2)} \quad (1)$$

$$Y = \frac{(-a2c2 + a1c2)}{(a2b1 - a1b2)} \quad (2)$$

Where, $a_1 = 2(x_2 - x_1)$; $b_1 = 2(y_2 - y_1)$; $c_1 = x_1^2 - x_2^2 + y_1^2 - y_2^2 - (d_1^2 - d_2^2)$;
$a_2 = 2(x_3 - x_2)$; $b_2 = 2(y_3 - y_2)$; and $c_2 = x_2^2 - x_3^2 + y_2^2 - y_3^2 - (d_2^2 - d_3^2)$;
Also note that,
$x_1 \neq x_2; x_2 \neq x_3$; and $y_1 \neq y_2; y_2 \neq y_3$

To calculate the accurate position of the target, localization is required. We use triangulation method to localize the target in 2-dimenstional space. For localization process three sensors are required whose locations $((x_1, y_1); (x_2, y_2); (x_3, y_3))$ are known to each other. When the target is detected then the sensors exchange their readings with each other (until the future-prediction algorithm is applied, sensors will exchange their readings to their 1-hop neighboring sensor nodes). At the time of triangulation the distance between the target and the 1st, 2nd and 3rd sensor is d_1, d_2 and d_3 respectively. To calculate the target's location (X; Y) at time instant 't', we apply the triangulation method and get Equations (1) and (2).

4.4. TARGET PREDICTION

The main theme of YAP essentially relies on predicting the location, and activating the most relevant sensor nodes for tracking the target. Two imminent elements for tacking the trajectory of a mobile target include the speed and bendability, the later being a measure of the yaw rate. In this section, we present mechanisms for determining both of these through the cooperation of sensors in the RoI.
The yaw plane translational equation is given as [14]:

$$r = \frac{F}{m.V} - \frac{d\beta}{dt} \quad \text{(I)}$$

Table 2: Description of equation (I)

Variable	Description
R	Yaw rate
F	Side force
V	Speed of the mobile target
M	Mass of the mobile target
β	Side slip angle
t	Instantaneous time

The initial side slip angle of the mobile target may be obviated. This causes an angular error in the initial measurements of the yaw rate. However, as the object moves, the error may be subsequently adjusted. Afterwards, the equation is,

$$r \Rightarrow \frac{F}{m.V} \qquad \text{(II)}$$

Considering m to be a constant of proportionality and can be obtained by target classification. Classification of the target is out of the scope of this paper. Many approaches are proposed for target classification such as K-nearest neighbor, maximum likelihood and support vector machine as described in [12]. Rest of the two factors in equation (II) are to be calculated i.e. F and V in order to determine the yaw rate.

Figure 4 describes the phenomena of yaw rate in our algorithm. As shown in Figure 4, arcs I, II, III represent the active regions, at time-stamps TS3, TS4 and TS5 respectively. The target tracking process starts at TS1, and the target prediction process starts at TS2. This is due to the fact that we cannot measure the target speed at TS1; the speed can only be measured once the target is moved to the next location. For our first prediction (as shown by the red dot), there is no drift error; so the yaw angle (α) was kept smaller. Yaw angle shows that how much target can deviate from its original path, so based on this concept we activate the sensors, which are closer to the yaw angle at both directions i.e. +α, –α. When the target physically moves to the next location at time TS3, the deviation of the target from the predicted location is calculated. So based on this drift from the prediction, different yaw angle is selected for the next prediction at TS4. It can be seen from the arc II that the yaw angle increases for the next prediction (as shown by the blue dot), because the error is greater. So the sensors which are closer to the yaw angle, at both ends i.e. (+α, –α), and near to the predicted location are activated. When the target moves and its actual location is measured at TS4, the error between actual and predicted location is calculated. As it can be seen from the figure 4, that the error is reduced from the previous one, so for the next prediction the yaw angle was kept smaller and the sensors closer to that yaw angle and to the predicted location(as shown by the grey dot) are activated, as shown in arc III.

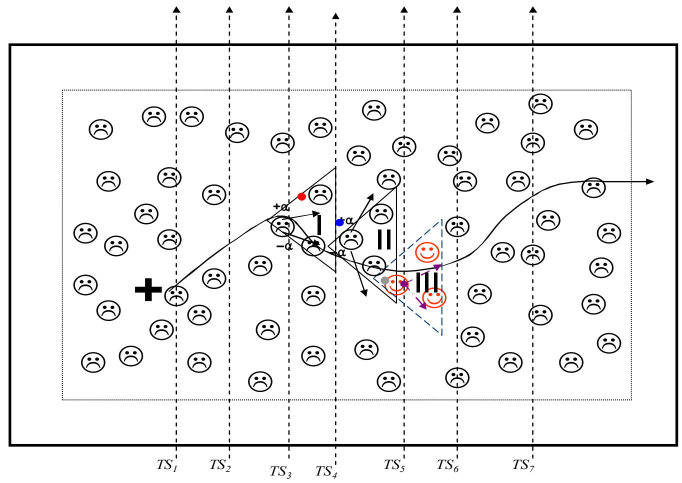

Figure 4: Speed and Yaw rate measurement

Based on yaw rate concept, the future target position is to be predicted by the algorithm that we described below. The flowchart of different activities is the best way to understand the overall picture about the sensor nodes activation and for the target prediction. For this reason, the flow chart is explained first, as shown in Figure 5

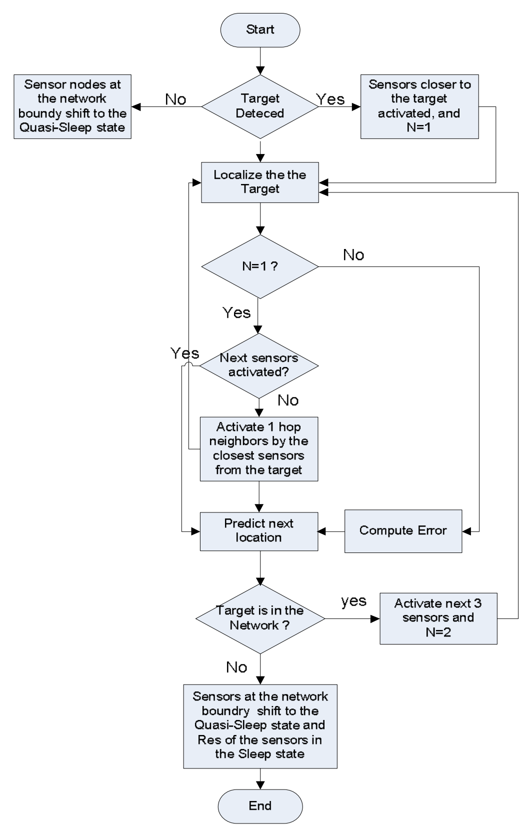

Figure 5: Flow chart of the Algorithm

Sensors become active from quasi-sleep mode in the presence of the target. Sensor nodes calculate the current location of the target through localization algorithm and the closest sensor to the target, broadcasts its reading to its 1 hop neighbors (at this point, speed 'S' can't be measured). When the target reaches closer to the next hop sensor nodes then the three sensor nodes, which are closer to the target, triangulate the target and measure the instantaneous speed, S, of the target as given in Equation (3).

$$S = \frac{CurrentTarget\ Location(x;\ y) - PreviousTarget\ Location(x;\ y)}{CurrentTime - TimeStamp('t')} \quad (3)$$

```
Legend:
    ETE:           External event occurs, which shows the presence of the target
    SNS:           Sensor states, which can be sleep state, quasi-sleep state or active
                   state
    LZ (T, X, Y):  It's a function which localizes the target and puts the x , y
                   coordinates of the target into X and Y
    PKT:           Packet (Target location  X , Y + timestamp' t_n')
    MLC:           Multicast Packet (Target location + timestamp' t_n')
    Δd:            Difference between current and previous target's location,
                   calculated as  (X1 - X) + (Y2 - Y)
    Δt:            Difference between current and previous time instant, is (t_n – t_(n–1))
    SP (d, t):     It's a function which takes inputs as Δd and Δt , and calculates
                   instantaneous speed of the target
    PR (d, X, Y, E):   It's a function which predicts next location of the target i.e.
                   ((Add Δd into X1, Y1) + Error) and overwrites values into X, Y
    ERR (A, P):    It's a function, which calculates the difference between target's
                   actual and predicted location, and computed as (Actual location –
                   Predicted location)
    N1:            The most nearest sensor node with respect to the predicted location
    N2:            The sensor node at angle +α with respect to the predicted location
    N3:            The sensor node at angle –α with respect to the predicted location

Begin Proc
  If ((ETE! = 0) && (SNS = = Sleep))
    SNS=Active
    LZ (Target, X, Y) at ' t0'
    Broadcast PKT in one-hop
    Wait ' T' seconds
    LZ (Target, X1, Y1) at ' t1'
    SP=Calculate (Δd, Δt)
    While (Target in Network) for n > 1
      tn=t(n-1) && n = n+1
      PR(Δd, X1, Y1, ERR(Actual, Predicted))
      (N1 = =Active && N2 = = Active && N3 = = Active)
      Multicast PKT to N1 , N2 & N3
      LZ (Target, X, Y) at ' tn'
      ERR (Actual, Predicted)
    End While ()
End Proc
```

Now we apply our prediction algorithm to predict the future target position. To understand our prediction algorithm we draw two possible target paths as shown in Figure 6 (a and b) below:

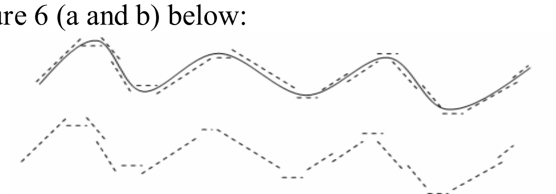

Figure 6 (a): Zig-Zag curve of the target path

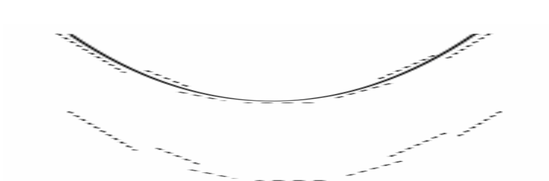

Figure 6 (b): Parabolic curve of the target path

Following observations are made from the Figure 6 (a and b)

- Every curve or complex path can be divided into smaller straight lines, whereas it can always be found that a line always passes through the two points [14] or we can say that combination of two points is called a line. And
- The length of straight line can vary depending upon the nature of the curve.

Hence we can say that every curve remains straight for some number of points depending upon the nature of the curve. So if we know the current location (x; y), speed 'S', and the yaw angle, then we can always calculate the next location with some error. Based on all these observations, we keep our prediction algorithm simple, and less computational. We say that for the first time, the next of the target is obtained by adding distance covered (from previous location to the current location), in the same direction i.e. straight path, but in reality target can deviate at any angle depending upon its speed [15], represented by yaw angle (α). So if the target deviates from its path that means there will be an error in our prediction. We calculate the error; this error also incorporates the side force affect on the target. So based on this error we predict the yaw angle (α), which will be greater for the high error rate and lesser for the low error rate. We say that target can deviate within the predicted yaw angle on both direction i.e. +α and –α. By keeping in view the goal of target tracking, where we need to activate minimum sensor nodes before the target is reached to the next region, we can activate three sensors as shown in Figure 4, such that 1st sensor (say N1) to the straight path as predicted, 2nd sensor (say N2) nearest at the angle +α and 3rd sensor (say N3) nearest at the angle –α. It means that even if there is an error, still the goal of target tracking is achieved; because the target will always lie within the yaw angle.

4.5. SUDDEN STOP

We can face uncertain behaviors of the target, at which the target reduces its speed suddenly, and then either it stops moving further or it gets turn to 180°. Under these conditions, our algorithm might not work, where the target may not lie within the predicted yaw angle. To address this problem, the concept of timer is introduced, where the activated sensors at target predicted location set a timer. The sensors wait for the target until the timer is expired. When the timer is expired, these sensors activate all other sensor nodes in their 1 hop neighborhood and report this "target lost" error back to the sensor node, which made the future prediction. On the reception of this packet, this sensor node also activates all other nodes in its 1-hop neighborhood. 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 those which localize the target. At this point, our algorithm will again take over and the target location prediction and tracking process resume.

5. RESULTS AND DISCUSSION

To elaborate our approach, we consider four paths of the target. To prove the validity of the algorithm, two paths are considered as more steeper than the other two, i.e. shown in Figure 7(b and c); Whereas the other two paths were considered as less steeper (which can be the case of high speed targets) as shown in Fig. 7(a and d). It can be seen from the Figure 7 that the target location prediction is very close to the actual one, this is due to the case that the rate of change of target speed is constant and also 'Δt' is kept smaller, where the target's bending nature can be measured at every time instant. It can be said that if 'Δt' is kept smaller, then the minimum will be the prediction error. Its is also observed that the prediction error in Figure 7(c) is more as compared to the one shown in Figure 7(a, b, d). This is due to the fact that target is changing its speed at every time instant 'Δt', also note that this differs from the practical cases in which targets maintain their speed for some amount of time 't', where 't' > 'Δt'. So it can be concluded as, if the steepness of the curve and the speed variation is more, the higher will be the prediction error. As discussed before, prediction is made and the three sensors are activated accordingly, even if the prediction is not accurate but still it is close enough to the actual position of the target where the target lies with-in the range of the three sensors, chosen by the yaw angle. Hence the goal of the target tracking can still be achieved with very optimal number of sensor nodes, even if the target shows uncertain behaviors.

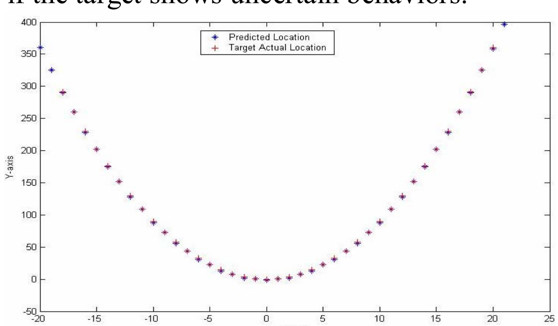

Figure 7(a): Comparison between the original and predicted path as parabolic behavior

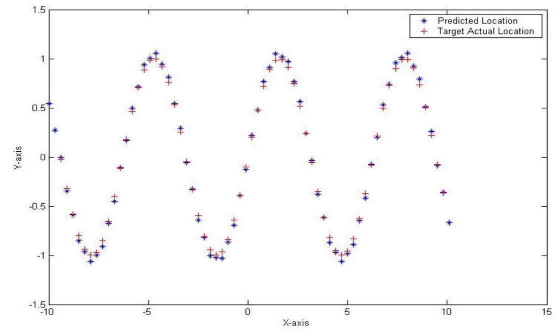

Figure 7(b): Comparison between the original and predicted path as Sine-wave behavior

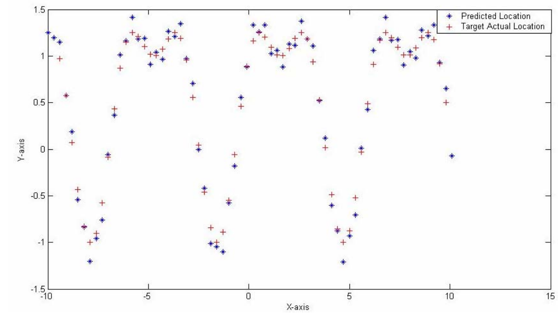

Figure 7(c): Comparison between the original and predicted path as general behavior

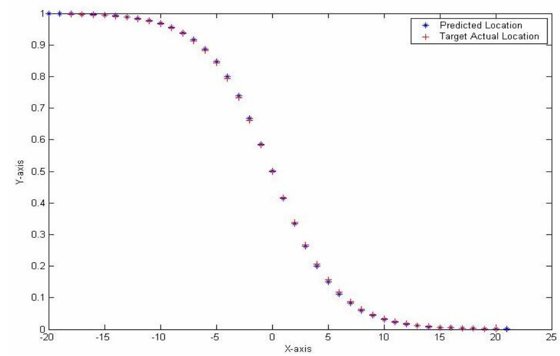

Figure 7(d): Comparison between the original and predicted path as general behavior

6. CONCLUSION

In this paper, a protocol is devised which is based on the phenomena of yaw rate. We build our network model, by discussing all the steps involved in the 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. Triangulation method is used for target localization. The YAP protocol is discussed in detail for target prediction, in which the future target prediction is based on the rate of change of the yaw angle, and on the target speed. To incorporate the freedom of movement of the target; yaw angle is incorporated, within which the target track can still be computed even if the predicted location is different than the actual one. We also discuss the case when the target is lost, due to the sudden stop behavior of the target. At the end, YAP is evaluated under different target's paths and results are discussed, which proves the accuracy of prediction for YAP in sensor networks.

ACKNOWLEDGEMENT

This work is supported by the IT R&D program of MIC/IITA, [2007-S-027-01, 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", *Proc. IEEE*, August 2003, pp: 1163- 1171

[5] H. T. Kung and D. Vlah, "Efficient Location Tracking Using Sensor Networks," *Proc., IEEE Wireless Communications and Networking Conference (WCNC '03)*, 2003.

[6] W. Zhang and G. Cao, "Optimizing Tree Reconfiguration for Mobile Target Tracking in Sensor Networks," *IEEE Transactions on Wireless Communication*, 2004.

[7] Francine Lalooses, Hengky Susanto, and Chorng Hwa Chang, "Recovery Target Tracking in Wildlife", *Electrical Engineering Department, Tufts University*, November 2005

[8] H. Yang and B. Sikdar. "A Protocol for Tracking Mobile Targets using Sensor Networks", *1st IEEE Workshop on Sensor Network Protocols and Applications*, pp. 71-81, 2003.

[9] Sundeep Pattem, 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*)

[10] S. Bandyopadhyay and E. Coyle. "Minimizing Communication Cost in Hierarchically Clustered Network of Wireless Sensors", *Purdue University*, 2004.

[11] Nadeem Ahmed, Salil S. Kanhere, Sanjay Jha, "The Holes Problem in Wireless Sensor Networks: A Survey," Mobile Computing and Communications Review, Volume 1, Number 2

[12] Dan Li, Wong K.D., Yu Hen Hu, Sayeed A.M., "Detection, Classification and Tracking of Targets in Sensor Networks", *Signal Processing Magazine, IEEE,* Mar 2002.

[13] Jason L. Hill, David E. Culler, "MICA: A Wireless Platform for Deeply Embedded Networks", Power consumption of mice hardware, *UC Berklay* (November-December 2002), 28 May 2007, http://www.cs.berkeley.edu/~culler/cs294-f03/papers/micaarch.pdf.

[14] Line (mathematics)- Formal Definition (17 May 2007), May 27, 2007; http://en.wikipedia.org/wiki/Line_(mathematics) Babister A W, "Aircraft Dynamic Stability and Response," Elsevier 1980, ISBN 0-08-024768-7

[15] P. T. Liu, "An optimum approach in target tracking with bearing measurements", *Journal of Optimization Theory and Applications*, vol. 56 and num. 2, pp: 205-214, 1988