Research Journal of Applied Sciences, Engineering and Technology 8(7): 892-899, 2014
DOI:10.19026/rjaset.8.1050
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: June 20, 2014 Accepted: July 13, 2014 Published: August 20, 2014

Research Article

Ant Based Compression and Sleep Scheduling Technique for Target Tracking In WSN

[1, 2]P.S. Velumani and [3]S. Murugappan
[1]Department of MCA, Model Institute of Engineering and Technology, Jammu and Kashmir, India
[2]M.G.R. Educational and Research Institute University,
[3]Department of Computer Science and Engineering, Annamalai University, Chennai, Tamilnadu, India

Abstract: The main challenges involved in target tracking applications of Wireless Sensor Networks (WSN) are energy consumption, accuracy and security. In order to reduce the amount of energy consumed in target tracking, in this study, Ant based compression and Sleep Scheduling technique is proposed. In this approach initially the information of the nodes are localized and tracked through ACO technique. After the localization process, the data which has to be transmitted to the destination node is compressed in order to make the network more efficient. A Sensor Node Activation Mechanism is developed, where the sensor nodes are turned into sleep mode for a specific time interval. Simulation results show that the proposed technique considerably reduces the network energy consumption.

Keywords: ACO, data compression, sensor node mechanism, WSN

INTRODUCTION

Wireless sensor network: Wireless Sensor Network (WSN) has become a very important topic with the rapid development that is vulnerable to a wide range of attacks due to deployment in the hostile environment. A WSN is a large network of resource-constrained sensor nodes with multiple preset functions, such as sensing and processing with number of low-cost, resource limited sensor nodes to sense important data related to environment and to transmit it to sink node that provides gateway functionality to another network, or an access point for human interface. These sensor networks are composed of energy constrained nodes embedding limited transmission, processing and sensing capabilities. Therefore network lifecycle becomes short and hence energy-efficient technique implementation becomes an important requirement for WSN.

The technology has improved significantly due to the usage of small, inexpensive, low-power, distributed devices. This has the potential of local processing and wireless communication which helps in the technological improvement. These are called as sensor nodes. By limiting the amount of processing, we restrict the ability of each sensor node. The physical environment can be measured in detail using the sensor nodes, where large number of other nodes co-ordinate to give the information. Thus the sensor network is a collection of sensor nodes which coordinate to perform some specific action. Dense deployment and coordination which are mainly used in the sensor networks differentiate them from the traditional networks. Industrial, military, biomedical and environment application areas widely use sensor networks. The sensor network applications include.

Intrusion detection and tracking: The battlefields are equipped with sensors along their borders so that they can detect, classify and track intruding personnel and vehicles.

Environmental monitoring: Specialized sensor nodes can detect temperature changes and smoke. These sensor nodes are arranged in high risk areas of a forest in order to provide early warning about forest fires.

Indoor surveillance: These surveillance sensor networks can be used for security purposes in an art gallery, or in a shopping mall.

Traffic analysis: Traffic sensor networks are used to monitor vehicle traffic on a highway or a crowded part of the city (Megerian and Potkonjak, 2002; Stankovic, 2006; Lewis, 2004; Akyildiz *et al*., 2002).

Localization in wireless sensor network: Location awareness is important for wireless sensor networks since many applications such as environment monitoring, vehicle tracking and mapping depend on

Corresponding Author: P.S. Velumani, Department of MCA, Model Institute of Engineering and Technology, Jammu and Kashmir, India

knowing the locations of sensor nodes. The localization problem has received considerable attention in the past, as many applications need to know where objects or persons are and hence various location services have been created. Undoubtedly, the Global Positioning System (GPS) is the most well-known location service in use today. Since most applications depend on a successful localization, i.e., to compute their positions in some fixed coordinate system, it is of great importance to design efficient localization algorithms (Mao *et al*., 2007; Amundson and Koutsoukos, 2009; Srinivasan and Wu, 2007; Kucuk *et al*., 2008).

Issues of localization: Self-localization capability is a highly desirable characteristic of wireless sensor networks. Sensor network localization algorithms estimate the locations of sensors with initially unknown location information by using knowledge of the absolute positions of a few sensors and inter-sensor measurements such as distance and bearing measurements. During this localization all the nodes in the network will be kept in active mode. Due to this the network's lifetime and also the nodes lifetime will be decreased. During the localization, the nodes in the network collect the data about the network. In this collected data it is necessary to reduce the redundant data; because of increase in redundant data the network's performance will be reduced.

Previous work: Velumani and Murugappan (2012) have proposed a target tracking and localization technique based on the Ant colony optimization with the help of anchor nodes. Initially a set of anchor nodes are deployed using the distance between the nodes. The Ant based routing protocol is used to find the best route with the shortest hop distance by acquiring the proximity information between every pair of anchor nodes. When a sensor node detects the target, the localization process is carried out using the anchor nodes and the position of the target is tracked using ant agents and the gathered information is transmitted to the sink. This approach reduces the latency in detecting the anchor nodes and also reduces the total number of anchor nodes to be deployed.

Velumani and Murugappan (2013) have proposed a secure authentication technique for localization in WSN. Initially the position of sensor nodes is estimated using proximity distance map computation. The anchor nodes then generate a location based key pair for each sensor node. This ensures that the attackers cannot exploit the positions and location based keys of compromised nodes. Then the mutual authentication of neighbor nodes is performed based on location information.

In this study, an ant based data compression and sleep scheduling technique is proposed to reduce the energy consumption involved in target tracking.

LITERATURE REVIEW

Li and Ren (2011) have proposed data compressing based target tracking protocol, which incorporates data approximation algorithm in the procedure of targets tracking. The characteristic of sensed data over sensor nodes surrounding interested targets is exploited, replaced as a series of approximate values. Compact descriptions of these readings are transmitted to the base-station, where targets location is implemented on the compact descriptions directly. Given an error bound, they have tried to compress readings for the same target maximally by grouping the original data and approximating them falls within the error bound around estimated values. The proposed approach can release the traffic load and reduce energy consumption for data transmission efficiently. The advantage of this approach is that proposed method yields excellent performance in energy savings and tracking quality.

Samarah *et al*. (2011) have proposed a Prediction based tracking Technique using Sequential Patterns (PTSPs) designed to achieve significant reductions in the energy dissipated by the OTSNs while maintaining acceptable missing rate levels. PTSP is tested against basic tracking techniques to determine the appropriateness of PTSP under various circumstances. The advantage of this approach is that the PTSP outperforms all the other basic tracking techniques and exhibits significant amounts of savings in terms of the entire network's energy consumption.

Dai *et al*. (2009) have proposed a new energy-efficient method based on Distributed Incremental Gene Expression Programming to discover the moving patterns of moving objects in order to turn on/off some sensor nodes at certain time to save energy. The main contributions include:

- Distributed GEP methods are used to perform collaborative mining the patterns of moving objects.
- Adjustable sliding window are adopted to balance the trade-off of the high accuracy and low energy consumption.

The advantage of this approach is that GEP-based motion prediction algorithm can greatly improve the tracking efficiency, increase the lifetime of the network by around 25% compared to other tracking algorithms, i.e., EKF and ECPA.

Soe (2008) has proposed a model to identify the lifetime of target tracking wireless sensor network. This model is static cluster based architecture and aims to provide two factors. First, it is to increase the lifetime of target tracking wireless sensor network. Secondly, it is to enable good localization result with low energy consumption for each sensor in the network. The model consists of heterogeneous sensors and each sensing member node in a cluster uses two operation modes-active mode and sleep mode. The advantage of this approach is that the architecture consumes less energy and increases lifetime than centralized and dynamic clustering architectures, for target tracking sensor network.

PROPOSED SOLUTION

Overview: As an extension to the previous works, we adopt an energy efficient approach for the localization. Also an approach is proposed to reduce the redundant data about the location information.

Initially in this approach the data about the nodes position in the network is gathered by the using an Ant Colony Optimization (ACO) technique of the previous study. To reduce the redundancy in the collected data, we apply data compression technique (Li and Ren, 2011). In this technique it actually compresses data from its member nodes as a compact structure with three entries mean, bitmap and variance. Finally to make the network more energy efficient, we propose to use the Sensor Node Activation Mechanism of Samarah *et al*. (2011). The main objective of this technique is to keep the set of nodes in the sleep mode for longest possible period for which no object is moving in their detection area.

The advantage of this approach is that the nodes in the network will be kept in sleep mode through which the life time of the network and also the nodes lifetime will be prolonged. Also by the data compression it is possible to reduce the redundant data present in the network which is collected by the nodes.

ACO: Initially, an Ant Colony Optimization (ACO) technique is used for the route discovery process. Each object creates a set of routing agents called FANT (Forward ANT) and Backward Ants (BANT) to search for the route to anchor node. In route discovery mechanism, the source will disseminate FANT to all its one-hop neighbors, when a path to the anchor node is to be established. While the anchor node is still not found, the neighbor would keep forwarding the FANTs to their own neighbors and so on. At the end of this route discovery mechanism, the routing table for each node contains the optimum path between the selected anchor nodes.

The routing protocol used for route discovery is concerned with finding the best route using ant colony optimization. In order to increase the coverage of the network, the shortest hop distance is calculated by acquiring the proximity information between every pair anchor nodes.

After calculating the shortest path distance using the ANT agents, all the anchor nodes gets the information about the position estimates of the each other. MDS provides a configuration about the anchor nodes.

Thus the proximity-distance map T among the anchor nodes can be calculated immediately.

Proximity Distance Map (PDM) calculation: We create two matrix tables: Proximity matrix contain the information about the hop count between the anchors. Geographical distance matrix contains the position estimates obtained from the Proximity matrix.

Let F be the proximity matrix that fxy is the proximity measure between anchors x and y where fxy = 0. The proximity measure can be hop count or cumulative path distance between anchors. Similarly, let G be a geographical distance matrix that gxy is the geographical distance between anchors x and y. gxy can be calculated by utilizing the position estimates obtained from the previous step. F and G are square matrices with size v * v where v is the total number of anchors.

The PDM J is the linear mapping that maps matrix F to matrix G and the following error is minimized:

$$Q = \| fiJ-ri F\|2 \quad (1)$$

where, fi = [fi1,. . . . fim] J and ri is the i-th row of J:

$$J = FGJ\ (GGJ)\ -1 \quad (2)$$

Localization of normal nodes: Each normal Sensor node (S) uses the mapping J to process the proximity vector vs. it has stored when it aided anchors exchanging proximity information:

$$X_S = Jvs \quad (3)$$

Finally, the node position is calculated by multi alteration with the processed proximity vector and the position information of primary and secondary anchors.

Data compression: We have assumed that each cluster head has m members, where the cluster member has information of proxy matrix (F). These nodes of the cluster are sorted according to the node id in the member list that is stored over the cluster head and base-station respectively. The data that is received from the member node can be represented as below.

$NF_X = \{(n_1, f_{X1}) ,\dots, (n_j, f_{Xj}), \dots, (n_m, f_{Xm})\}$, where n_j is the node id and j is the serial number of node n_j in member list, $j \in (1, m)$. X_j is a d-dimensional vector $(x_{1j}, \dots, x_{kj},\dots, x_{dj})$, where d is the number of targets in the monitoring area and x_{kj} is the reading of target k, $k \in (1, d)$. If the cluster head does not receive data of target k from node n_j, it sets x_{kj}-1

The cluster head then compresses the data received from its member nodes as a compact structure with three entries.

Mean (M): It defines the mean value data received from all member nodes.

Bitmap (B): It is a map that indicates whether the sensed data of a sensor node can be approximated to mean within a given compressing error bound ε. The i^{th} bit (i = 1, 2…) is used to indicate whether the data from the i^{th} node can be approximated by mean or not and if it can be approximated then we set the value of bit to 1,

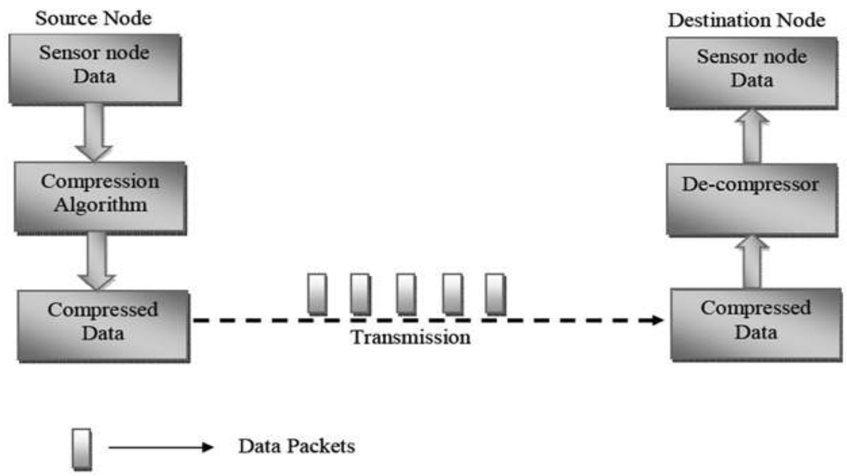

Fig. 1: Data compression

else we will keep it as 0. For example, if a cluster has four member nodes and their sensed data is 30, 32, 30 and 34, respectively. Then their expected mean value will be 31.5. If the given compressing error is 2, then bitmap will be 1 i.e., 0001 in binary. As the first three data falls within ε around mean, it will be replaced by mean and as the last value falls outside ε of mean, the 4th bit of bitmap is updated to 1.

Variance (V): It is defined as an active array to store the variance of data when it falls more than the specified error constraint ε away from mean.

After time T_w, the cluster head set bitmap as 0. To simplify the presentation, we use $b_1b_2 \ldots b_m$ to represent the bits of bitmap, where $bi \in \{0, 1\}$, it is the value of i^{th} bit of bitmap.

As the compressing mechanism is applied independently for each target, so we will consider only the sensed data of target k for simplicity. First step includes calculating the expected value of all data in Nf_X and then the cluster head validates whether the data in Nf_X can be replaced by mean with guarantee of compressing error. For each data x_j^k, if it is -1, it means that the data of node n_j is not received, then set bi 1 and write 0 to variance. Otherwise, its variance of mean σ_{kj}^2 is calculated. If $\sigma_{kj}^2 \leq \varepsilon$ the cluster head replaces x_j^k by mean. Otherwise, set bi 1 and write σ_{kj} to variance sequentially. We observe that the number of 1 in bitmap shows the number the values have been written to variance.

Let the set of data be $Nf_X = \{(n_1, f_{X1}), \ldots, (n_j, f_{Xj}), \ldots, (n_m, f_{Xm})\}$ and these set data will be input for the algorithm and let ε be the compressing error bound ε. The output of this algorithm will be a compact data description:

```
Set bitmap = 0
Computer mean of all received data
For each data x_j^k in Nf_X
If (x_kj == -1)
Set b_j to 1
Append 0 to variance [getIndex (j)]
Else
: Compute its variance of mean σ_kj^2
: If (σ_kj^2>ε)
: Set b_i to 1
: Append σ_kj to variance [getIndex (j)]
: End if
: End if
: End for
: Return a compact structure <mean, bitmap, variance>
```

The above algorithm describes the compressing algorithm. Here the cluster head computes the expected value of all received data for the same target and assigns it to mean. For each data x_{kj}, if it is within ε of mean, then it can be filtered out, else the j^{th} bit of bitmap is set 1, meanwhile the variance of x_j^k and ε is written to variance [getIndex (j)]. getIndex () is a function that returns the corresponding subscript of σ_{kj} in variance by counting the number of 1 in $b_1, \ldots, b_m$.

In the Fig. 1 during the transmission of the data packets the source node compresses the data packets and then send it to the destination node. This compression of data packets are done using an energy efficient algorithm above. Through that compression algorithm the data packets are compressed. These compressed data packets are then sent to the destination node. At the destination node, the node receives the compressed data packets. The compressed data packets decompressed at the destination node. Through this way of data compression the energy of the network can be efficiently utilized.

Sensor node activation mechanism: After the process of localization and the data compression of the nodes and the data packets in the network. The network tries to maximize the energy efficiency of the network. To maximize the energy efficiency the network keeps the nodes in the sleep mode which are not active in the

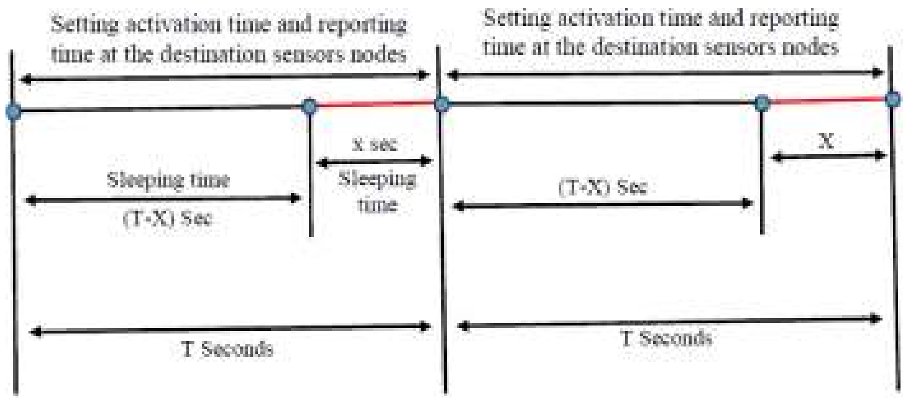

Fig. 2: Node activation mechanism

network. The main goal of this stage is to keep in sleep mode for the longest possible period for any sensor node that has no object moving in its direction area and hence saving its energy. Moreover, in case of a moving object in the vicinity of a certain sensor node, the sensor node will not be awake all the time. It must switch to sleep mode as long as possible when it is not impairing the tracking process and this sensor node will be called as current sensor. The current sensor switches to active mode for X ms and during this time, the current sensor senses its detection area. It informs all the finding to the base station by the end of X that also matches the end of T (the time period between reports). Figure 2 shows the relation between X, T and (T-X). The last action performed by current sensor before switching to sleep mode is predicting the location of moving object during the (T-X) milliseconds and sending a message to the destination sensors to wake up after the (T-X). The current sensor makes use of the embedded sequential patterns to decide which destination sensor is to be activated. The factors that need to be considered while applying sequential pattern(s) on a particular sensor is based on the minimum confidence level required by the application that would be required from any sequential pattern before it could be used to predict the future movements of a moving object. Hence, if the application sets the minimum confidence level as 70% then any sequential pattern with a confidence level lower than 70% will not be executed. Determining the most appropriate minimum confidence level requires a thorough analysis of prediction model performance, that will be discussed in future research.

Both the current sensor and the destination sensor get activated after passing of (T-X) ms and then it starts tracking the moving object. Moreover, if any one of the active sensor nodes detects the object in its vicinity, then it will send an Acknowledgement message (ACK) to the former current sensor, unless it is the current sensor. This action is performed by the sensor node to become the new current sensor so that it may start the previously illustrated process. But, the rest of the active sensor nodes will be in sleep mode.

Algorithm: In this approach the network uses an energy efficient approach in order to utilize the network's energy in an efficient manner. Here the network localizes and targets the destination, compresses the data packets during the transmission and uses a mechanism to send the node to sleep:

Step 1: Initially the network localizes and targets the destination node using the ACO approach of the previous work. This ACO approach consists of Proximity Distance Map (PDM) calculation and Localization of Normal Nodes.

Step 2: After the localization of the node the cluster head compresses the data packets. The cluster head compresses the data packets received from its member nodes as a compact structure with three entries mean, bitmap and variance.

Step 3: Finally after the localization of nodes and the compression of the data packets in the network. The network does this to maximize the energy efficiency the network keeps the nodes in the sleep mode which are not active in the network. The node will be sent to the sleep mode for particular time period and a message about that particular node will be sent to the destination node.

SIMULATION RESULTS

Simulation parameters: The Ant based Compression and Sleep Scheduling technique for target tracking (ACSS) technique is evaluated through Network Simulator (NS2) (http:///www.isi.edu/nsnam/ns). We use a bounded region of 500×500 sqm, in which 100 sensor nodes are randomly placed and a sink node is located in the center of the network. We have used two target objects which randomly move across the network, whose location information has to be tracked by the sensor nodes. The power levels of the nodes are assigned such that the transmission range and the sensing range of the nodes are all varied from 250 to

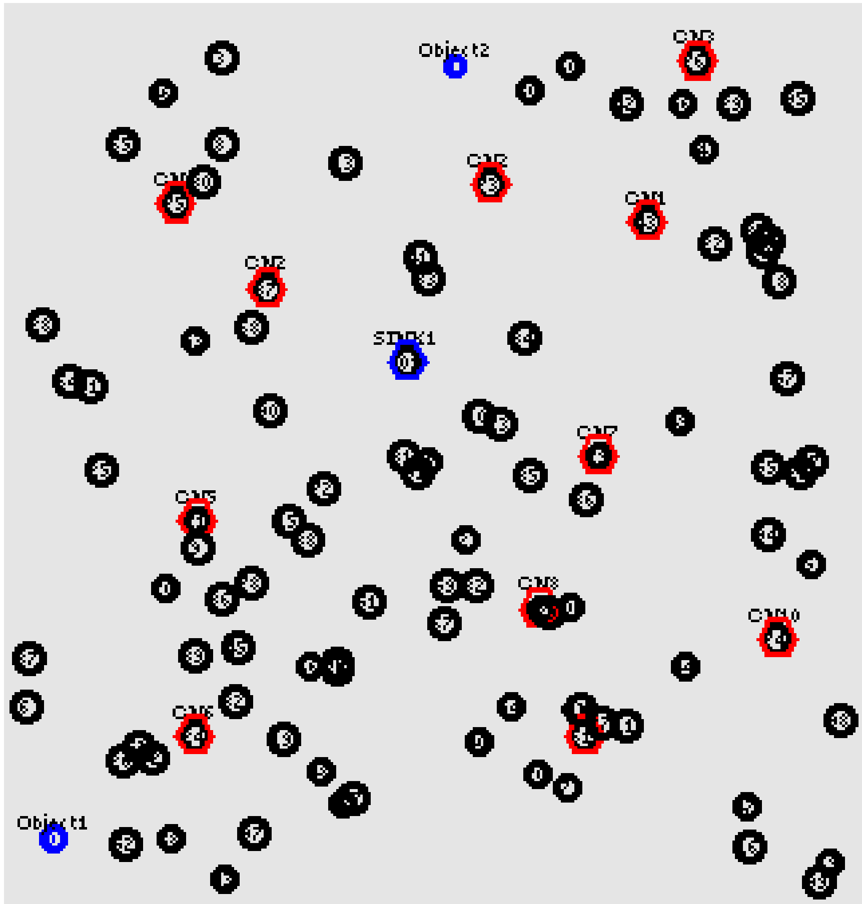

Fig. 3: Simulation topology

Table 1: Simulation settings

No. of nodes	100
Area size	500×500
Mac	802.11
Simulation time	50 sec
Traffic source	CBR
Packet size	512
Transmit power	0.360 w
Receiving power	0.395 w
Idle power	0.335 w
Initial energy	12.1 J
Transmission range	250, 300, 350, 400 and 450 m
Routing protocol	ACSS
Speed	5 m/sec
Rate	100, 200, 300, 400 and 500 Kb

450 m. In the simulation, the channel capacity of mobile hosts is set to the same value: 2 Mbps.

We vary the transmission range as 250, 300, 350 and 450 m and transmission rate as 100, 200, 300, 400 and 500 Kb, respectively.

The simulated traffic is Constant Bit Rate (CBR). The simulation topology is given in Fig. 3.

The Table 1 summarizes the simulation parameters used.

Performance metrics: The performance of our proposed ACSS technique is compared with the ATTL [ ] method. We evaluate mainly the performance according to the following metrics.

Average energy consumption: The average energy consumed by the nodes in receiving and sending the packets.

Packet delivery ratio: It is defined as the number of data packets received successfully with the total number of packets sent.

Average end-to-end delay: It includes the localization delay, tracking delay and transmission delay.

Results:

Based on transmission range: In our first experiment we vary the transmission range as 250, 300, 350, 400 and 450 m, respectively.

Figure 4 shows the end-to-end delay of ACSS and ATTL techniques for different range scenario. We can conclude that the delay of our proposed ACSS approach has 6% of less than ATTL approach.

Figure 5 shows the delivery ratio of ACSS and ATTL techniques for different range scenario. We can

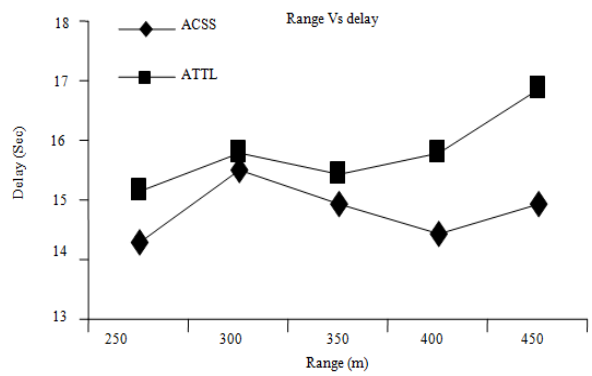

Fig. 4: Range vs. delay

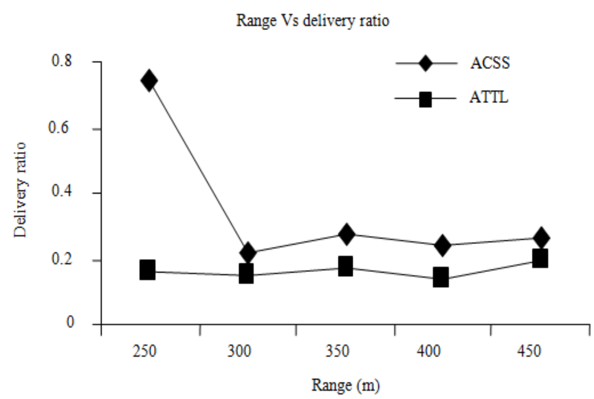

Fig. 5: Range vs. delivery ratio

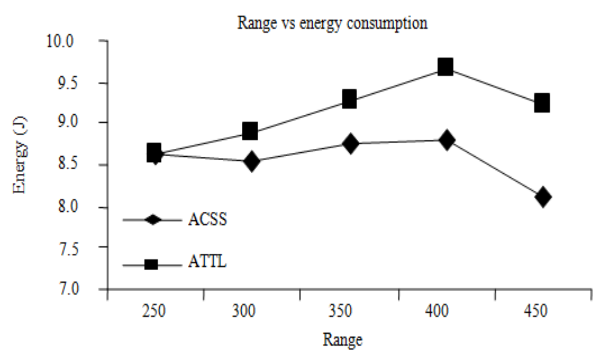

Fig. 6: Range vs. energy consumption

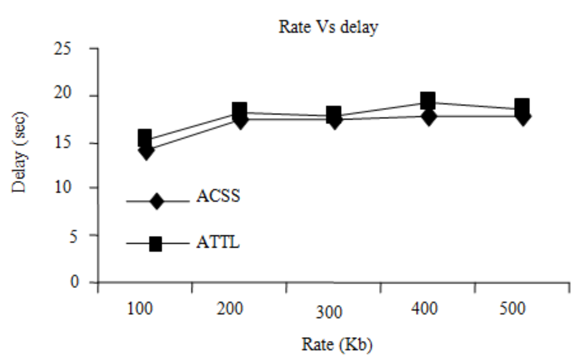

Fig. 7: Rate vs. delay

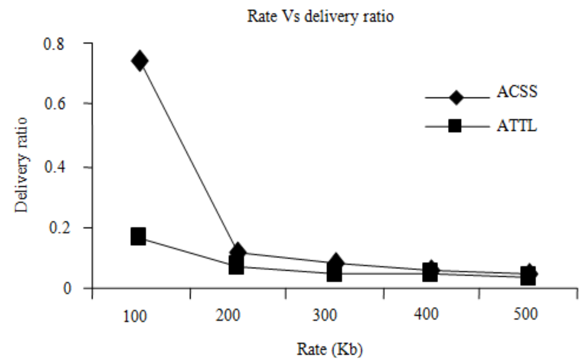

Fig. 8: Rate vs. delivery ratio

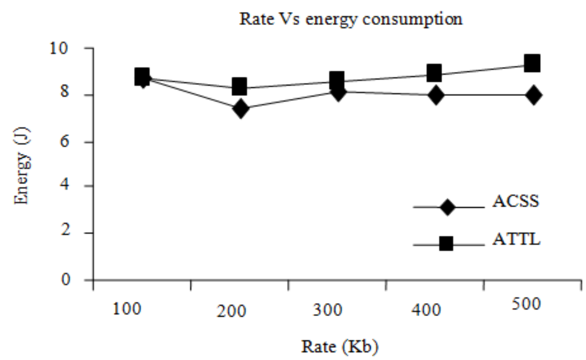

Fig. 9: Rate vs. energy consumption

conclude that the delivery ratio of our proposed ACSS approach has 43% of higher than ATTL approach.

Figure 6 shows the energy consumption of ACSS and ATTL techniques for different range scenario. We can conclude that the energy consumption of our proposed ACSS approach has 6% of less than ATTL approach.

Based on rate: In our second experiment we vary the transmission rate as 100, 200, 300, 400 and 500 Kb, respectively.

Figure 7 shows the end-to-end delay of ACSS and ATTL techniques for different rate scenario. We can conclude that the delay of our proposed ACSS approach has 4% of less than ATTL approach.

Figure 8 shows the delivery ratio of ACSS and ATTL techniques for different rate scenario. We can conclude that the delivery ratio of our proposed ACSS approach has 44% of higher than ATTL approach.

Figure 9 shows the energy consumption of ACSS and ATTL techniques for different rate scenario. We can conclude that the energy consumption of our proposed ACSS approach has 8% of less than ATTL approach.

CONCLUSION

To overcome the issues in target tracking in sensor nodes, a compression and scheduling technique has been proposed. ACO technique is used for localization and tracking. The tracked data is then compressed and sent to the sink. In order to save the energy consumption, the nodes are periodically made to enter into sleep mode by means of sensor node activation mechanism. By simulation results, it has been shown that the energy consumption was reduced and packet delivery ratio was increased by the proposed technique.

REFERENCES

Akyildiz, I.F., W. Su, Y. Sankarasubramaniam and E. Cayirci, 2002. Wireless sensor networks: A survey. Comput. Netw., 38(4): 393-422.

Amundson, I and X.D. Koutsoukos, 2009. A survey on localization for mobile wireless sensor networks. In: Fuller, R. and X.D. Koutsoukos (Eds.), 2nd International Workshop on Mobile Entity Localization and Tracking in GPS-less Environnments. Springer, Berlin, Heidelberg, 5801: 235-254.

Dai, S., C. Tang, S. Qiao, Y. Wang, H. Li and C. Li, 2009. An energy-efficient tracking algorithm based on gene expression programming in wireless sensor networks. Proceeding of the 1st International Conference on Information Science and Engineering (ICISE, 2009), pp: 774-777.

Kucuk, K., A. Kavak, H. Yigit and C. Ozdemir, 2008. A novel localization technique for wireless sensor networks using adaptive antenna arrays. Proceeding of the Radio and Wireless Symposium, pp: 483-486.

Lewis, F.L., 2004. Smart Environments: Technologies, Protocols and Applications. Wiley, New York.

Li, J. and Q. Ren, 2011. Compressing information of target tracking in wireless sensor networks. Lect. Notes Comput. Sc., 3(2): 73-81.

Mao, G., B. Fidan and B. Anderson, 2007. Wireless sensor network localization techniques. Comput. Netw., 51(10): 2529-2553.

Megerian, S. and M. Potkonjak, 2002. Wireless Sensor Networks. Encyclopedia of Telecommunications, Wiley Publishers, pp: 1-13.

Samarah, S., M. Al-Hajri and A. Boukerche, 2011. A predictive energy-efficient technique to support object-tracking sensor networks. IEEE T. Veh. Technol., 60(2).

Soe, K.T., 2008. Increasing lifetime of target tracking wireless sensor networks. Proceeding of the World Academy of Science, Engineering and Technology. Vol. 32.

Srinivasan, A. and J. Wu, 2007. A survey on secure localization in wireless sensor networks. In: Furht, B. (Ed.), Encyclopedia of Wireless and Mobile Communications, CRC Press, Taylor and Francis Group.

Stankovic, J.A., 2006. Wireless Sensor Networks. Department of Computer Science, University of Virginia Charlottesville, Virginia.

Velumani, P.S. and S. Murugappan, 2012. Ant based target tracking and localization technique for wireless sensor networks. Eur. J. Sci. Res., 70(4): 554-568.

Velumani, P.S. and S. Murugappan, 2013. Secure authentication technique for localization in wireless sensor networks. Int. Rev. Comput. Softw., 8(9): 2239.