An Innovative Approach for Solving Clock Drift Management Problem using Differential Evolution Algorithm

Nibha Tiwari, Sheela Verma

Abstract- Like distributed systems, wireless sensors networks often require a synchronization of time for consistency and coordination of data. Time synchronization is a critical piece of infrastructure in any distributed system. Time synchronization is mostly required in wireless sensor network. Having time synchronization in sensor network it allows collective signal processing, and helps in efficient sharing of the communication channel. Time synchronization is a critical piece of infrastructure for any distributed system. Distributed, wireless sensor networks make extensive use of synchronized time, but often have unique requirements in the scope, lifetime, and precision of the synchronization achieved, as well as the time and energy required to achieve it. The protocol which was developed for time synchronization of wireless sensor networks was Flooding Time Synchronization Protocol (FTSP) In this paper, FTSP is taken under the consideration for clock drift management using differential evolution (DE) algorithm. The paper calculates clock skew and the clock offset, generates linear line and optimizes the value of average time synchronization error using Genetic and DE algorithm. This paper dictates implementation and experimental results that produce reduced average time synchronization error optimized using DE, compared to that of linear regression used in FTSP.

Keywords- Differential evolution (DE), Flooding Time Synchronization Protocol (FTSP).Time Synchronization, Average Time Synchronization Error, Clock Drift, Wireless sensor network.

I. INTRODUCTION

Recent advances in miniaturization and low-cost, low power design have led to active research in large-scale, highly distributed systems of small, wireless, low-power, unattended sensors and actuators. Recent advances in micro-electromechanical (MEMS) technology have led to the development of small, low-cost, and low-power sensors. Wireless sensor networks (WSNs) are large-scale networks of such sensors, dedicated to observing and monitoring various aspects of the physical world. In such networks, data from each sensor is agglomerated using *data fusion* to form a single meaningful result, which makes time synchronization between sensors highly desirable. Many synchronization algorithms have been proposed for WSN [11]. In the Timing-sync Protocol for Sensor Networks (TPSN) [3] hierarchical structure is used to synchronize the whole WSN to a single time server. TPSN requires the root node to synchronize all or parts of the nodes in the sensor field. It consists of two phases: (1) the level discovery phase, where the hierarchical structure is built in the network starting from the root node; and (2) the synchronization phase, where pair wise synchronization is performed throughout the network. In the Reference-Broadcast Synchronization (RBS) [12], a reference message is broadcasted. The set of receivers that are within the reference broadcast of a sender synchronize with each other. It is a receiver–receiver synchronization method, eliminates sender side uncertainty. This decreases the synchronization error and improves the efficiency. RBS is applicable to any medium which has broadcast capabilities including wired and wireless networks. For many applications of wireless sensor networks, such as data gathering, it is of utmost importance to know when data was sampled or when a given event happened. Therefore, sensor nodes need to have a common understanding of time. Synchronized clocks can also help in other applications such as the development of more energy efficient MAC protocols.

Revised Manuscript Received on 30 October 2012.
* Correspondence Author
Nibha Tiwari, M.Tech. Scholar, Department of Computer & Engineering, SSEC, Bhilai (C.G.), India.
Sheela Verma, Asst. Processor, Department of Computer & Engineering, SSCET, Bhilai (C.G.), India.

II. DIFFERENTIAL EVOLUTION

Differential Evolution is Stochastic, population-based optimization algorithm. It was Developed to optimize real parameter, real valued functions. It initializes a randomly generated population vector when no preliminary knowledge about the solution space is available. The classical version of DE is “DE/rand/1/bin”[4][5] and mostly used.
Let the solution vectors in generation *G are*
$\boldsymbol{P_{nj}Q(j = 1, 2, \ldots N_p)}$
where N_p is the population size.

Here with the help of DE we will generate trial vectors. New trial vectors will be generated by using various types of Crossover operators and Mutation, and which of the vectors will be successfully selected into the next generation will be determined by selection.

Mutation

A mutant vector $\boldsymbol{M_i,Q}$, For each target vector = $\boldsymbol{P_{j,Q}}$,j=1,2,3…..N_P a mutant vector is generated according to the formula

$$M_{j,Q+1} = P_{r1},Q + F\,(P_{2},Q - P_{r3},Q)$$

Where r1,r2,r3 are the random indexes belong to1,2,3….. N_P and are mutually different and F>0.

Crossover

In DE crossover operator is used to build trial vectors by recombining two different vectors. It is defined as follows:

$$c_{j,Q+1} = (c_{1j,Q+1},\ c_{2j,Q+1}, \ldots\ldots\ldots\ldots\ldots\ldots\ c_{dj,Q+1}\ )$$

is formed ,where

$$C_{ij,Q+1} = \begin{cases} M_{ij,Q+1} & \text{if } random_i(0,1)<CR \\ P_{ij,Q} & \text{otherwise} \end{cases}$$

i=1,2……….d

CR is the crossover constant [0:1]] which has to be determined by the user, and $random_i$ is the randomly chosen index1,2,………..d.

Selection

To decide which vector (C_{jQ} , P_{jQ}) should be a member of next (new) generation *G+1*,we use selection. The vector which is giving the higher fitness value is chosen.

III. PROPOSED APPROACH

In this paper the variants of operators of DE are used with FTSP for clock drift management. The cycle of time synchronization is initiated by the single, node which is dynamically elected root. We will try to coordinate the whole network with the time of a single node i.e. the root. The root node sends beacons to other nodes which are local nodes having clock drift for estimation of root's clock global time The beacons contain global time at fixed intervals of time (say T).

We can break the time synchronization into following events.

- The process gets started by the root node. The root node transmits the beacon having its timestamp i e global time (GT).The time stamping is done before broadcasting the packets.
- When a beacon is received by any collector node it marks its own local time(LT) in the packet just when the radio receives the packet. The synchronization point is determined by the pair of global-local time and is used to calculate the offset using
  Offset =GT-LT

Table 1 in section 4 is created by time synchronization module, a table of global-local time pair of each local node for every fixed interval T. Table 2 in section 4 gives the offset of each node at every fixed interval T. These GT-LT pair points of a node at each time interval T are used to estimate the drift of that particular local node with respect to root node.

3.1clock drift management using linear regression

With the help of synchronization points we have drawn the linear regression line.Using linear regression method clock skew (C_s)and initial offset(I_{offs}) is calculated between global and local node. With time synchronization error, local node can estimate the global time using linear regression line.

$$C_s = \frac{Ns*\sum GT*Offs-(\sum GT)*(\sum Offs)}{N_s*\sum GT^2-(\sum GT)^2}$$

$$\text{Initial offset } (I_{offs}) = \overline{Offs} - C_s*\overline{GT}$$

$$\text{Where } \overline{Offs} = \frac{\sum Offs}{N_s}$$

$$\text{And } \overline{GT} = \frac{\sum GT}{N_s}$$

N_s= number of times beacons are send

Hence equation of linear regression line generated is:

$$\mathbf{y = mx + c.}$$

3.2 optimization of clock drift using DE

In FTSP linear regression is used to calculate the linear line which gives the synchronized time or offset. We have used DE to synchronize nodes. To use any optimization algorithm, We have to first formulate the optimization algorithm. Each individual of DE represents the solution. In FTSP the solution is the offset which represents the local time of each node in synchronized manner. Each individual solution of DE represents the offset used in calculation of synchronized times.

Dimension of a solution is equal to the no. of offset or the no. of times the message is sent between the nodes. In linear regression the approximation line near the offset value are generated.DE generates different points near offset values and by combining these points the line is drawn. The line containing point's gives minimum error is considered as a solution.

$$\mathbf{F=\sum_{j=1}^{n} Offset_i-X_{ij}}$$

Here $offset_i$ is the original offset of wireless node with index j.

x_{ij} is the i^{th} individual of DE with j points or dimension value.

3.3algorithm

```
while m=0 to the number of  population do
{
   while n=0 to the dimension of   individual do
   {
          Initialize the individuals   randomly within the
          space of search
   }
   For each individual evaluate   average time
     synchronization error.
   Compute the best fitness value based on the above
     equation.
}
 for Mutation Phase  do
  {
     Generate mutant vector by using   DE mutation.
  }
 for Crossover Phase do
while (m=0 to number of population)
{
  while (n = 0 to number of   dimension)
   {
     If crossover criteria met
      Than Perform crossover
}
Result of  Crossover is the New candidate solution generated.
Generate best fitness value  of candidate solution .
For Selection phase do
```

```
  {
        On the basic of fitness value, select a solution
        If candidate solution has a    better   fitness value
        than replace old solution
        Calculate best solution
  }
}
until (maximum number of iteration)
```

IV. RESULTS

The algorithm given in section 3.3 is implemented on visual c++. The algorithm is tested for different pairs of (root-local) nodes. The root node is selected randomly. We have assumed node number 2 is root node, local node 3 is estimating global time. The root node sends beacons at an interval of 100 units (T). The number of times (n) the beacon message is sent is taken to 5. Suppose 5 nodes are in network and 2nd node is selected as root node then result pair comprises of 2-1, 2-3, 2-4, 2-5. Results are shown in next section for one pair i.e. 2-3 as one pair only is sufficient to proof that using DE average time synchronization error is optimized for reduced value.

Table 1. Time stamping for global and local node at intervals of 100 units of time And Offset calculated at local node. (Synchronization Point Table)

Global time and Offset calculated using DE		
Node	Global time (Y)	Offset (X)
1	156	81.997
2	256	89.32
3	356	94.66
4	456	103.99
5	556	111.32

	Global and Local Times of wireless Sensor Node	
Node	Global time	Offset
1	156	85
2	256	86
3	356	94
4	456	104
5	556	112

The root node transmits the beacons marked with global timestamp at an interval of 100 units and local node having clock drift receives it and puts its own local time stamp shown in Table 1.

The offset calculated at local node for each send of beacon shown in the table.

4.1 Result of clock drift management using linear regression.

Figure 1 shows a linear regression line is generated using the synchronization points of table 2. The average time synchronization error calculated using equation (8) is 2.08

Table:2

Global time and Offset Calculated using Linear Regression		
Node	Global Time (y)	Offset (x)
1	156	81.792
2	256	89.962
3	356	96.156
4	456	103.352
5	556	110.562

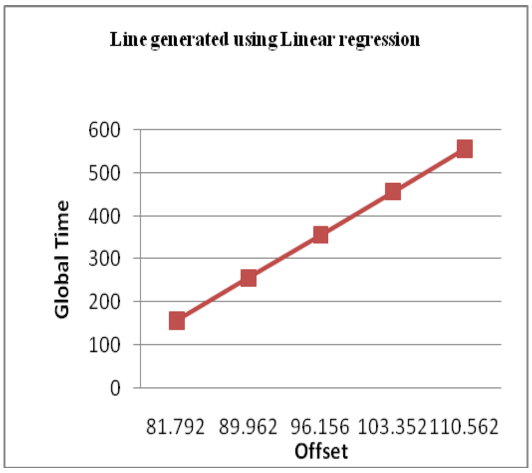

Fig. 1. Linear line generated using linear regression.

4.2 result of clock drift management using deferential evaluation

DE is run for each pair after time synchronization cycle get completed described in section 3. The parameters for the algorithms are standard. Some of the parameters which have been varied are:

- The number of iterations is 50.
- The number of times (N) the beacon message is sent by root node is 5.

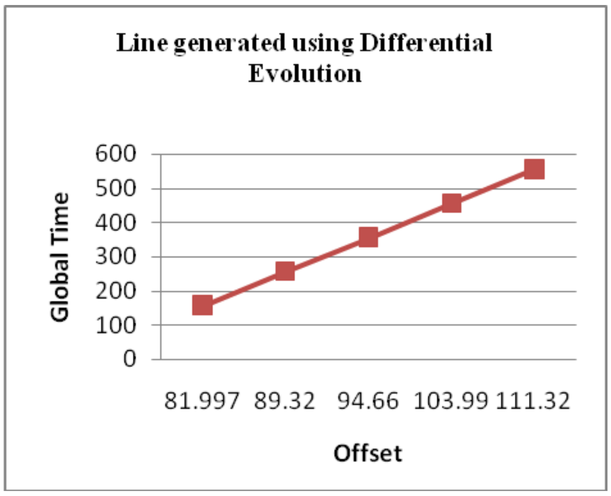

Fig. 2. Linear line generated using DE.

Figure 2 show the linear line generated using DE. The average time synchronization error using DE is **1.93368**.

4.3 Comparative result of differential evaluation with linear regression.

Average Time Synchronization Error using Linear Regression and DE depicts the average time synchronization error using linear regression and DE for different number of nodes in network. It can be seen by using DE average time synchronization is considerably reduced.

V. CONCLUSION

We require time Synchronization in a Wireless Sensor Networks (WSN) to allow proper correlation of diverse measurements and for an efficient sharing of the communication channel. This research estimates the drift of the receiver clock with respect to the sender clock, optimizes the value of average time synchronization error using Differential evaluation (DE). The algorithm is implemented on Visual C++ platform. The average time synchronization error calculated by DE is compared with the value of average time synchronization error calculated by linear regression. Using DE average time synchronization error is reduced to 1.93368 compared to linear regression.

REFERENCES

1. Maroti, M., Kusy, B., Simon, G., Ledeczi, A.: Flooding time synchronization in wireless sensor networks. ACM SenSys'04, Baltinore, Maryland, pp. (2004).
2. Elson, J. E. *Time Synchronization in Wireless Sensor Networks. Ph.D. Thesis, University of California, Los* Angeles 2003.
3. Ganeriwal, S., Kumar, R., and Srivastava, M. B. *Timing-Sync Protocol for Sensor Networks. The First ACM Conference* on Embedded Networked Sensor System (SenSys), p. 138–149, November 2003.
4. Price, K. and Storn, R. (1996), Minimizing the Real Functions of the ICEC'96 contest by Differential Evolution, *IEEE International Conference on Evolutionary Computation (ICEC'96)* may 1996,
5. Storn, R. and Price, K., Differential Evolu tion a simple and efficient adaptive scheme for global optimization over continuous spaces,Technical Report TR-95-012, ICSI, http://http.icsi.berkeley.edu/-storn/litera.html
6. R.C. chakraborty "Fundamentals of genetic algorithms (2010)
7. Josef Tvrdik University of Ostrava, Department of Computer Science" Adaptive Differential Evolution and Exponential Crossover".
8. Bharath Sundararaman, Ugo Buy, and Ajay D. Kshemkalyani" Clock Synchronization for Wireless Sensor Networks: A Survey " March 22, 2005.
9. Jeremy Eric Elso " Time Synchronization in Wireless Sensor Networks"2003.
10. 1]Maroti, M., Kusy, B., Simon, G., Ledeczi, A.: Flooding time synchronization in wireless sensor networks. ACM SenSys'04, Baltinore, Maryland, pp. (2004)
11. Stojmenovic, I.:HandbookOf Sensor Networks Algorithms and Architectures. Canada : John Wiley & Sons (2005)
12. Elson, J. E., Girod, L., Estrin, D.: Fine-Grained Network Time Synchronization using Reference Broadcasts. The Fifth Symposium on Operating Systems Design.