www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 3 March 2015, Page No. 10760-10763

EMU-Synchronization Enhanced Mobile Underwater Networks for Assisting Time Synchronization Scheme in Sensors

K.Priyanga,V.Radhika,L.D.R.Vinoodune,Mrs.Josepha Menandas M.E.,
sripriyanga24@gmail.com, radhikavidyashankar@gmail.com, vinoodune913@gmail.com, josepha82@gmail.com
Department of Computer Science and Engineering,
Panimalar Engineering College, Chennai.

***Abstract*—Efficient data transmission is a critical issue for wireless sensor networks (WSNs). Clustering is an effective and practical way to enhance the system performance of WSNs. In this paper, we propose a cluster-based synchronization algorithm for underwater acoustic sensor networks based on MU-Sync, called EMU-Sync. In underwater sensor networks the time synchronization is challenging for long propagation delay, sensor node mobility and energy consumption. The improved performance of MU-Sync from the EMU-Sync gives the ability to maintain its low Complexity. This is possible by allowing the cluster head to calculate the skew and the offset from the view of both a cluster head and a neighboring node. Simulation results confirms that EMU-Sync offers better performances than MU-Sync in both accuracy and energy efficiency.**

Keywords—Time Synchronization, Underwater acoustic sensor networks, Mobility, Sensor networks,

I. INTRODUCTION

Data mining generally categories into various function such as classification, clustering, searching. Classification in data mining is termed as collection of target categories/classes. Classification is mainly used for portioning the data in to different classes. Generally the term classification referred as the process of generalizing the data according to different instances.

The main aim is to predict the target class accurately. Classification predicts the target class accurately .classification are discrete and it doesn't have any order. The classification algorithm finds relationships between the values of the predictors and the values of the target. Different algorithm uses different technique to find these relationships.

After classifying the data the datasets are need to be grouped in to a single domain. The process of grouping data is called as clustering. The data objects that are similar will be in a same group and those are dissimilar will be in a different group.

Cluster analysis is not an automatic task but it is an iterative process. There are various classification models which uses different types of clustering models which uses different types of clustering methods. For example connectivity model uses hierarchical clustering which is mainly based on distance. There are various types of clustering methods such as

1. Strict portioning clusters
2. Strict portioning clusters with outliers
3. Overlapping clustering
4. Hierarchical clustering
5. Subspace clustering

The data mining involves various common tasks such as anomaly detection, association rule learning, clustering, classification, regression and summarization. Given a set of words of length up to n. The set of words are classified and clustered .The data can be retrieved by means of a searching process. By using the search process the best match for the given input string can be find.

The approximate membership Extraction (AME) is a dictionary based entity search process. It takes more searching time and it causes many redundancies. To overcome this problem approximate membership localization is proposed.

II. LITERATURE SURVEY

Any device with clock system may not provide the actual time correctly due to errors. It leads to the needs of time synchronization. To obtain a perfect time synchronization in mobile underwater, there are two main challenges. The first challenge is long and dynamic propagation delay that makes the calculation of time propagation delay become difficult. Second, since the nodes rely on battery power for operation, the synchronization should consume low energy.

A. Long & Dynamic Propagation Delay

Most of acoustic underwater time synchronization algorithms utilize the technique of two-way message exchange in the message delivery time which causes several uncertainties such as sending time, access time, propagation time, receive time, etc. In these uncertainties, propagation delay is the major barrier because of low speed of acoustic underwater signal and the motion of nodes in the water. For these reasons, there is no way to find out real propagation delay hence the error still exists.

B. Energy consumption

process. Hence, a simple communication procedure, i.e. small number of message exchanges is required to reduce the energy consumption

III RELATED WORK

In recent years, there is growing interest in time synchronization for underwater wireless sensor networks. However, the research is still limited. From literatures, TSHL [3] is designed to estimate skew and offset by using one-way and two-way communications respectively for the high latency networks. However, a common assumption of the constant propagation delay during the message exchanges in static networks is not applicable in mobile networks. MU-Sync [4] is a cluster-based protocol, in which the cluster head is responsible for starting the time synchronization process and for calculating the skew and offset for all nodes within the cluster. MU-Sync performs twice linear regression. For the first linear regression , the cluster head estimates skew to reduce the effect of skew during the processing time of the neighboring node. For the Second linear regression, the skew and offset are estimated. Although, MU-Sync is designed to solve the long and dynamic propagation delay, the calculation of propagation delay from half of the round trip time is inaccurate.

Mobi-Sync [5] is different from the previous methods. The Mobi-Sync structure consists of three types of nodes, namely surface buoy, super node and ordinary node. The surface buoys are equipped with GPS to obtain the global time. The super nodes are assume to be able to communicate with surface buoys in real time. In practice, this assumption is not realistic. The ordinary nodes will synchronize with the super nodes by spatial correlation of velocity of the super nodes. To achieve good time synchronization, it required minimum three or more super nodes. D-Sync [6] and DA-Sync [7] utilize the Doppler shift to estimate velocity of the nodes. In D-Sync, the estimated velocity is used for estimating the propagation delay. Since there is error in the estimated velocity, it will certainly result in error in the estimation of the propagation delay. On the other hand, DA-Sync the estimated velocities are leveraging by Kalman filter before used in the propagation delay estimation. However, the leveraging process requires a good precision in the velocity measurement. This is difficult to archive.

Although, Mobi-Sync, D-Sync and DA-Sync are more efficient than MU-Sync, they require complex computation and have their own limitation. On the other hand, MU-Sync is simpler and require less computation. All of the contents above leads to the proposed algorithm called "Enhance MU-Sync (EMU-Sync)" which is a cluster based synchronization algorithm for underwater acoustic sensor networks based on MU-Sync. The design algorithm reduces the error of skew and offset by calculating both cluster head and neighboring node.

IV EMU-SYNC

The EMU-Sync (Enhanced MU-Sync) is an improved MU-Sync protocol. As stated in Section III that MU-Sync is a simple but low accuracy protocol. This inaccuracy is mainly caused by the assumption that the one-way propagation delay of each direction during the message (a REF packet) exchange are the same which is rarely true for mobile underwater network. As a result, the estimation of skew and offset of

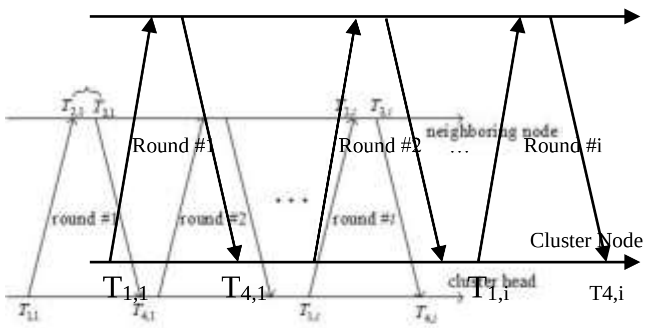

Figure. 1: Message Exchange

MU-Sync is only correct for the following conditions: 1) when the cluster head is static while the neighboring node can be static or mobile, 2) Both cluster head and neighboring node are mobile in the same speed and direction. For other cases, cluster head is mobile and neighboring node is static and both neighboring node and cluster head are mobile in different speed and different direction, the estimation of skew and offset is incorrect since estimation of the propagation delay from half of the round trip time is incorrect.

EMU-Sync can alleviate the above-mentioned problem of MU-Sync by calculating the skew and offset by averaging the estimated skew and the estimated offset from both neighboring node and cluster head at cluster head side. The estimation error can be reduced by taking the average of the estimated skew and offset.

In general, time synchronization use two parameters namely skew and offset as equation

$$T = at + b, \quad (1)$$

where T , t, a and b are local time, global time, skew and offset respectively.

The EMU-Sync is designed to solve the problem from the worst case by calculating skew and offset of both the cluster head and neighbor node at cluster head. As in step 1 of Fig. (1), in step 1 the cluster head sends a synchronization message at time T_1 ,then a neighboring node receives the synchronization message at time T_2. In step 2 a neighboring node sends the synchronization message at time T_3, then the cluster head receives the synchronization message at time T_4 and repeat the same procedure for n round. Time stamp of each node is its local clock that can be expressed as equation (1).

$$T_1 = a_c t_1 + b_c, \quad (2)$$

$$T_2 = a_n t_2 + b_n, \quad (3)$$

$$T_3 = a_n t_3 + b_n, \quad (4)$$

$$T_4 = a_c t_4 + b_c, \quad (5)$$

where a_c and b_c are the skew and the offset of cluster head while a_n and b_n are the skew and the offset of the neighboring node, respectively. The global time, t_2 and t_4 can be represented as equation (6) and (7)

$$t_2 = t_1 + d^{c\to n}, \quad (6)$$

$$t_4 = t_3 + d^{n\to c}, \quad (7)$$

where $d^{c\to n}$ and $d^{c\to n}$ are the propagation delay from a cluster head to a neighboring node and from a neighboring node to a cluster head, respectively. Since the propagation delay from step 1 and step 2 are unknown and unequal, we assume the propagation delay can calculate from half of the round trip time in each round as

$$d_i^{c\to n} = d_i^{n\to c} = \frac{\left(T_{4,i} - T_{1,i} + \frac{(T_{2,i} - T_{3,i})}{\hat{a}}\right)}{2}, \quad (8)$$

where i denotes the message exchange round number. Sim-ilarly to MU-Sync, $\hat{a}$ is the skew estimation obtained from the first linear regression with Least Mean Square (LMS) operation. To reduce the effect of node's mobility, the prop-agation delays obtained from (8) are subtracted from the $T_{2,i}$ and $T_{4,i}$. The cluster head then applies second linear regressions over the data points obtained from the previous step. Instead of performing a second linear regression over the data points ($T_{1,i}$, $T_{2,i}$) to obtain the estimated skew and offset of the neighboring node based on a perspective of a cluster head, EMU-Sync performs linear regressions to obtain the estimated skew and offset of the neighboring node based on the perspective of both a cluster head and a neighboring node which are denoted as $\hat{a}_{n,c}$, $\hat{b}_{n,c}$, $\hat{a}_{c,n}$ and $\hat{b}_{c,n}$, respectively.

To reduce the effect of the assumption $d_i^{c\to n} = d_i^{n\to c}$ in MU-Sync, we obtain the final estimated skew and offset by

$$\hat{a}_n = \frac{(\hat{a}_{n,c} + \frac{1}{\hat{a}_{c,n}})}{2}, \quad (9)$$

$$\hat{b}_n = \frac{(\hat{b}_{n,c} - \hat{b}_{c,n})}{2}, \quad (10)$$

where $\hat{a}_n$ and $\hat{b}_n$ are the average estimate skew and offset respectively. Finally, cluster head broadcasts these value to its neighboring nodes so that each can keep itself synchronized with each others.

V SIMULATION RESULTS

A. Simulation setup

In our simulation, the nodes are placed randomly according to uniform distribution and are allowed to move randomly within an area of 1000 x 1000 m. The movement model of a node is the same as the one used in [4]. The speed of sound underwater is assumed to be constant at 1500 m/s and there is no skew variation and no packet collision during message exchanges. As suggested in [8], the non-deterministic errors are modeled using Gaussian distribution, with a receive jitter of 15µs. Unless specified otherwise, the following set of parameters are used in the simulations:

- Clock skew is 50 ppm.
- Clock offset is 800 ppm.
- The duration a node takes before responding to a REF packets (T_W) is 0 s.
- Maximum speed of a sensor node (V_{max}) is 2 m/s.
- The number of REF packets used to perform linear regression is 25.
- The time interval between two successive REF packet is 5 s.
- Clock granularity is 1µs.

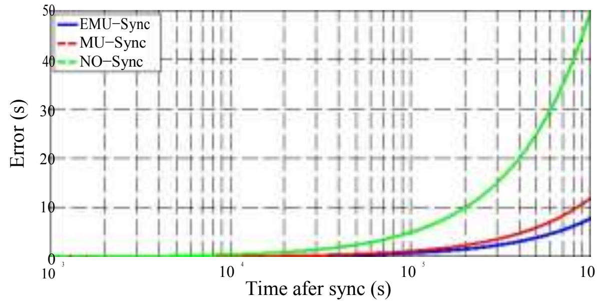

Figure. 2: The error in time estimate VS the time elapsed since synchronization.

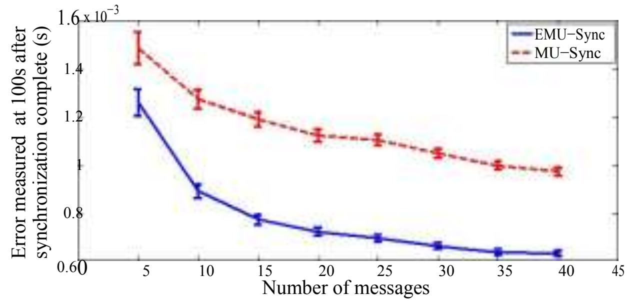

Figure. 3: Effect of changing the number of messages

B. Results

Each data point shown in the simulation results is obtained from the average of 10,000 simulation runs. The error bar associated with each data point represents the standard devia-tion. Note that the term “No-Sync” indicates the performance of a node that do not apply any synchronization scheme. As a result, the performance of No-Sync is expected to be the worst among the studied schemes (e.g., EMU-Sync, MU-Sync and No-Sync). Fig. 2 shows that the synchronization error keeps increasing as time goes by for all schemes. However, the performance of EMU-Sync is better than MU-Sync while No-Sync performs the worst as expected. This performance improvement of EMU-Sync, when compared with MU-Sync, confirms that the one-way propagation delay esti-mation method proposed in EMU-Sync yields higher accuracy than the one used by MU-Sync.

Figure. 3 indicates that for the same number of control

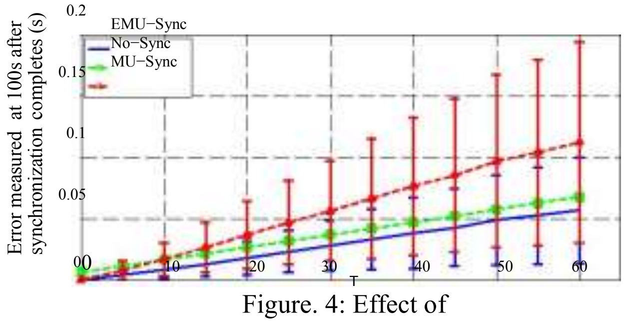

Figure. 4: Effect of T_W.

TABLE I: Table of frequency of re-synchronization with vary process time and error tolerance

Protocols	T_w	e	$\hat{a}_n$	$a - \hat{a}_n$	$b - \hat{b}_n$	κ
MU-Sync	0	0.01	56.9 ppm	6.9 ppm	169 ppm	607
		0.05				120
	5	0.01	98 ppm	48 ppm	5713 ppm	9720
		0.05				941
EMU-Sync	0	0.01	54.1 ppm	4.1 ppm	40 ppm	356
		0.05				71
	5	0.01	84 ppm	34 ppm	2498 ppm	3910
		0.05				617

messages used during the linear regression process, to obtain both the estimated skew and offset, EMU-Sync can achieve significant lower error than MU-Sync. This implies that in order to achieve the same performance, EMU-Sync requires lesser number of control message exchanges, making it a higher energy-efficient protocol.

Next, we examine the effect of waiting duration T_W on the performance of each scheme. From the results shown in Fig. 4, it is obvious that a large value of T_W leads to high synchronization error for both MU-Sync and EMU-Sync. Surprisingly, the performance of MU-Sync is so sensitive to T_W that its performance is worse than No-Sync when T_W is greater than 10 s. For the case of EMU-Sync, although its performance degrades with increasing T_W, it is still more robust than MU-Sync since it maintains significant better performance than both No-Sync and MU-Sync. The main reason causing MU-Sync to perform badly when T_W increases is due to the assumption of $d_i^{c\to n} = d_i^{n\to c}$ that leads to large error, especially in mobile network. To elaborate further, assuming the case that two nodes are moving at the same speed but with the opposite direction, the longer the neighboring node waits before responding to the cluster head, the larger the value $d_i^{c\to n} - d_i^{n\to c}$ as well as higher synchronization error. Although EMU-Sync also uses the half of a round trip time in calculating $d_i^{c\to n}$ and $d^{n\to c}$, averaging of the estimated skew and offset from the perspective of both cluster head and neighboring node before obtaining the final estimate skew and offset helps to minimize the error.

To understand the performance gain achieving from EMU-Sync over MU-Sync, we attempt to calculate the energy efficiency (ρ) for both protocols using:

where ␣ and γ are the number of message used in performing a linear regression and the REF packet size (in bytes), respectively. κ denotes the number of re-synchronization required within a certain duration, denoted as ϑ. TABLE I shows an example of how to obtain κ. Specifically, we run simulations to obtain the estimated skew (a_n) and offset (b_n) for T_W = 0 and 5 s and e = 0.01 and 0.05 s. Moreover, ϑ,␣ γ are set to 10 days, 25 and 32 bytes, respectively. These values are then used to calculate κ according to

$$\kappa = \frac{\vartheta}{\hat{\ } + (\hat{\ })_{a-\hat{a}}} \quad (12) \quad {}_{ae\ b-b}$$

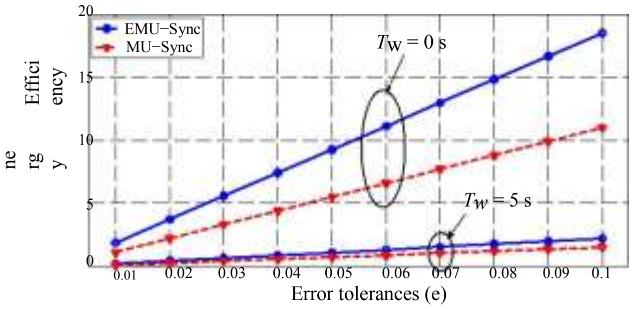

Figure. 5: Energy efficiency with varying error tolerance

κ is then used in (11) to obtain ρ which are shown in Fig.5. It is obvious that EMU-sync has a better energy efficiency than MU-Sync for all range of error tolerances, although the significance decreases with increasing T_W (generally, T_W <1 s).

VI. CONCLUSION

In this paper, we present EMU-Sync, a time synchroniza-tion protocol developed for mobile underwater network. The protocol is an enhancement of MU-Sync. By estimating the skew and offset of the node using an average between the estimated skew and offset of the node based on the perspective of both a cluster head and a neighboring node, EMU-Sync is able to show significant gain in both accuracy and en-ergy efficiency over MU-Sync. Despite this performance gain, EMU-Sync is able to maintain the attractive characteristics of being a simple and low complexity protocol of MU-Sync. Extensive simulation results also confirm that EMU-Sync is highly robust to the variation of the duration the node takes before responding to the REF packet to which MU-Sync is highly sensitive.

REFERENCES

[1] I. F. Akyildiz, D. Pompili, and T. Melodia. Challenges for efficient communication in underwater acoustic sensor networks. *ACM SIGBED Review*, 1(1):38, July 2004.

[2] I. F. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: Research challenges. *Ad Hoc Networks (Elsevier)*, 3(3):257279, March 2005.

[3] A. A. Syed and J. Heidemann, Time Synchronization for High Latency Acoustic Networks, in Proc. *INFOCOM 2006*, April 2006, p

[5] J. Liu, Z. Zhou, Z. Peng, J.-H. Cui, M. Zuba, and L. Fiondella. Mobi-sync: Efficient time synchronization for mobile underwater sensor networks. In *IEEE Transaction on Parallel and Distributed Systems (TPDS)*, Feb 2012.

[6] F. Lu, D. Mirza, and C. Schurgers. D-sync: Doppler-based time synchronization for mobile underwater sensor networks. *WUWNet10*,2010.

[7] J. Liu, Z. Wang, M. Zuba, Z. Peng, J.H. Cui and S. Zhou. DA-sync: A Doppler Assisted Time Synchronization Scheme for Mobile Under-water Sensor Network. In *IEEE Transaction on Mobile Computing*, 2014

[8] J. Elson, L. Girod, and D. Estrin, Fine-Grained Time Synchronization using Reference Broadcasts, in *Proc. 5th Symp. Op. Sys. Design and Implementation*, Boston, MA, Dec. 200