Efficient Time Synchronization in Wireless Sensor Networks

Shehzad Ashraf Ch., Aftab Ahmed Khan, Zahid Mehmood, Muhammad Ahsan Habib, Qasim Mehmood

***Abstract*—**Energy efficiency is the key requirement in wireless sensor network as sensors are small, cheap and are deployed in very large number in a large geographical area, so there is no question of replacing the batteries of the sensors once deployed. Different ways can be used for efficient energy transmission including Multi-Hop algorithms, collaborative communication, cooperative-communication, Beam- forming, routing algorithm, phase, frequency and time synchronization. The paper reviews the need for time synchronization and proposed a BFS based synchronization algorithm to achieve energy efficiency. The efficiency of our protocol has been tested and verified by simulation

***Keywords*—**time synchronization, sensor networks, energy efficiency, breadth first search

I. Introduction

TIME Synchronization in wireless networks is not only important for basic communication, but it also provides the ability to detect movement, location, and propinquity. The synchronization process can be as consisting of four parts the send time, access time, propagation time, and receive time.

Time synchronization is a critical piece of infrastructure for any distributed system. Distributed wireless sensor networks make extensive use of synchronized time: for example, to measure the time of flight of sound for localizing its source [11], to integrate a time-series of proximity detections into a velocity estimate [2] to distribute a beamforming array [12] or to suppress redundant messages of an event detected by different sensors in a network [6]. Most of the traditional distributed systems requirements are also the requirements of sensor networks, accurate timestamps are often needed in cryptographic schemes to verify the freshness of data received, to coordinate events scheduled in the future, for ordering logged events during system debugging, and so forth.

II. Literature Review

[2] Described the requirements in the scope, life time and precision of the synchronization achieved as well as time and energy required to achieve it. Different requirements affecting the sensor energy are precision, lifetime, scope and availability, efficiency, cost and form. All communication operation are energy extensive even passive listening have significant effects on energy reserves, so to save energy the sensor should be in sleep mode not in listening mode. An algorithm *post-facto* synchronization was proposed in [2] which was based on the above idea of sleeping the sensor till the generation of some event to be monitored, every sensor is equipped with a pre-processor which will wake the sensor up when some event occurred. Energy cost of message exchange is also high as compared to communication cost, reducing the average number of message exchanges can significantly reduce the energy consumption.[1]. [5] Proposed a clock synchronization algorithm in ad-hoc networks, the idea was to generate time stamps using unsynchronized local clocks the receiving sensor translate it to its local clock.

[6] Claimed to improve accuracy of time synchronization by exploiting global network wide constraints satisfied by the very notion of time. A distributed algorithm was proposed in [6] to achieve the synchronization through a completely asynchronous, the idea was to employ only local broadcasts. If O_{ij} is the offset of the clock at node j with respect to the clock at a neighbouring node i, at a certain time. An estimate of b O_{ij} can be formed by bilateral exchange of time stamped packets between the neighbouring nodes i and j. it also formed an arc matrix which describes the direction of synchronization. Due to unpredictability and imperfect measurability of message delays in a network environment, physical clock synchronization is always imperfect [3]. [5] proposed a model based algorithm for clock synchronization in networks with drifting clocks, A reach ability tree was designed keeping in mind the distances between the sensors, an estimate of drifting clocks was also drawn, it also consider the delays of communication for synchronization between two and more sensors .[3] proposed a hierarchical wireless sensor network, in which each sensor on lower layer will synchronize the sensor of upper layer, it was assumed that clock drifts between sensors is linear and sensors exchanges time stamps to estimate the best linear fit. Each sensor is capable of communicating directly to the sensors located at a distance less than the transmission radius [8], two algorithms TINY-SYNC and MINI-SYNC was also proposed, TINY-SYNC was used to synchronize two sensors directly accessible to each other while the purpose of MINI-SYNC was to synchronize the entire network. The scheme of [3] was having a lot of resemblances with scheme of [5]. [7] Proposed position based routing algorithm using DFS, it used an

Mr. Shehzad Ashraf Ch, Zahid Mehmood and Aftab Ahmed khan are associated with the International Islamic University Islamabad, Pakistan (e-mails shahzad@iiu.edu.pk, malik.aftabahmed@gmail.com, zahid@iiu.edu.pk,)

Mr. Ahsan Habib and Dr. Qasim Mehmood are with Iqra University, Islamabad, Pakistan (e-mail ahsanhabib77@yahoo.com, qasim@iqraisb.edu.pk)

optimized routing scheme having eliminated from the candidate list the neighbours whose messages to other sensors were overhead, a routing algorithm was proposed which was integrated with power metrics minimizing the total power for routing a message.

III. Proposed Architecture

Modified master slave architecture is being proposed in our solution, having a number of sensor nodes, a base station and some arbitrary event generator, the nodes are deployed randomly but are able to locate each other also sensor nodes and base station can locate each other. After deployment the base station will calculate and propagate the adjacency list of each sensor node. Proposed solution is using the *post-facto* synchronization [2], each sensor node consists of a pre-processor and a processor all the sensor nodes are in sleep mode (i.e. low energy mode having processor off and pre-processor on) till the happening of some arbitrary random event. When an event is occurred, the sensor node who first receives the event information will wake its processor on and will act as master node. The master node will be the initiator for time synchronization all other nodes can be termed as receptors. Initiator will broadcast the synchronization message to all its neighbour, the neighbours will propagate synchronization message to their neighbours and so on until the node which is closest to base station will receive synchronization information, it will also act as bridge to communicate between base station and sensor nodes. After synchronization all the nodes will start monitoring the event and the information will be sent aggregately to the base station through the bridge node.

IV. Network Structure

Sensor nodes can be deployed in two distinct ways Fixed array antenna and distributed wireless sensor networks, in fixed array antenna the distances between the base station and all the sensors are known well in advance and are equivalent to each other, so in case of fixed array antenna the synchronization process is of very low worth, because of equal distance the phase frequency offsets can be calculated easily also the noise can be estimated easily. But its not the real time scenario, the sensor are always deployed randomly (thrown by some airplane etc.), so the available real time sensor network termed as distributed sensor network is having distinct distances between base station and different sensor nodes.

The fixed size array and distributed sensor network are shown in figure 1 (a) and (b).

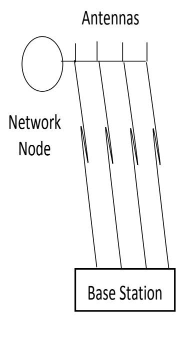

Fig. 1 (a) Fixed Antenna Array

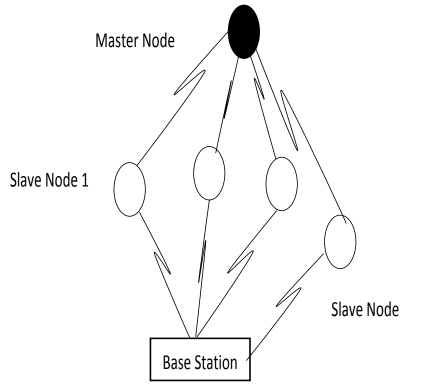

Fig. 1 (b) Distributed Sensor Network

Figure 2 is our proposed distributed modified master slave architecture having an event generator a master node, the slave nodes, a base station, some communication media/ internet and an information receiver and processor.

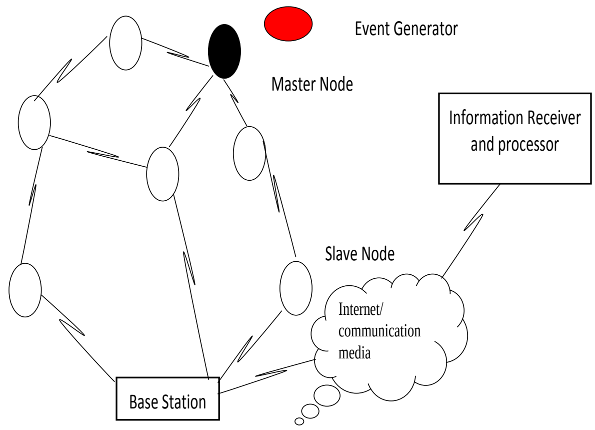

Fig. 2 Proposed Distributed Sensor Network

The sensor nodes can work as master or slave according to the requirements. A sensor node will become master if it receives event information directly from event generator. When event is generated the master node start synchronizing slave nodes by transmitting the sync signal to the slave nodes in its forward and backward Adjacency list**.** On reception of event information each node's pre-processor wakes its processor on then the event is monitored by that node and information is transmitted to all the nodes in its adjacency list for synchronization, In the same fashion all nodes are synchronised, channel coefficient are also estimated by each node and base station during synchronization to reduce error rates.

After synchronization the data is sent by master node to all adjacent slave nodes, the data received by each node is then retransmitted to all the nodes in the adjacency list of each slave node who received information by master node.

The Main goals of proposed solution are reducing the number of message passing for synchronization and Energy efficient monitoring of events by sensor nodes.

V. Synchronization Process

Time synchronization will start after happening of some event, Breadth first search method will be used for time synchronization, where the neighbours of the initiator node are synchronised first then the neighbours of the neighbours are synchronised and so on till the remotest neighbour is synchronised. The algorithm works as follows

```
BFS(G, s)
for each vertex u ∈ V[G] – { s }

    do state[u] ←READY
    state[s] ← WAIT
Q ←φ
ENQUEUE(Q,s)
while Q ≠ φ
   do u←DEQUEUE(Q)
        for each v  ∈ Adj[u]
            do if state[v] = READY
                then state[v]←WAIT
                ENQUEUE(Q, v)
state[u] ←PROCESSED
```

It marks all the nodes and creates a BFS-Adjacency matrix which is be propagated to all the nodes. When ever an event is generated by some event generator the master node initiates synchronization process an example is shown below

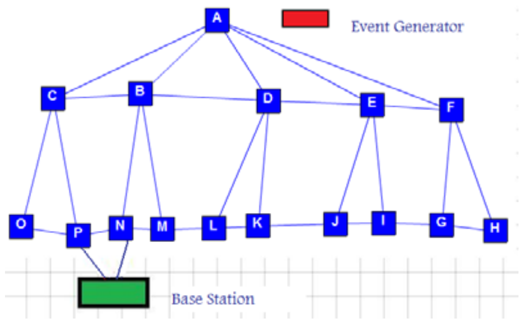

Fig. 3 A sample graph of sensor nodes, event generator and base station

In figure 3 The BFS algorithm is used for creating a spanning tree which is used as forward and backward adjacency list for time synchronization, An event is generated near node A, it will act as master node and will start propagating the event information to all nodes using time synchronization, time synchronization will be performed in a sequence A C B D E F G H I J K L M. It is a spanning tree and does not contain any cycle.

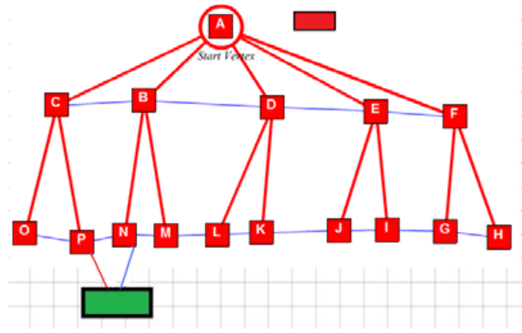

Fig. 4 BFS spanning tree for master node ***A***

The worst running time for BFS Time synchronization is

$$T_{BFS} = O(|V| + |E|)$$

Where V is number of sensor nodes and E is V×V, In worst case all nodes will have edges with each other so the worst running time for BFS is

$$T_{BFS} = O(|V^2|)$$

VI. Performance Evaluation

A customized simulator was implemented in visual studio.net to evaluate the performance of our proposed mechanism. The neighbors of a sensor node are determined manually by creating edges between the node and its neighbors. Breadth first search algorithm was tested on modified master slave architecture. Below are the comparisons of message passing between proposed solution and some other models, the results are as follows

TABLE I
COMPARISON OF NUMBER OF MESSAGE PASSING AND MAXIMUM HOP COUNTS

Scenario	**Messages Passing**	**Hop Count**
Sparse Network	nlgn	lgn
Mesh Network	n(n-1)+1	1
Ring Network	2(n-1)+1	n
Arbitrary Network	4n	lgn
Proposed Solution	n	lgn

Table I describes the number of message passing for synchronization process, In a mesh network where all the nodes are having edges to each other (total number of edges are n(n-1)) Number of messaging passing is directly proportional to n^2, where as that of sparse network message passing is directly proportional to nlgn, the ring to 2n and the arbitrary to 4n, while the proposed solution remains proportional to n. The hop count complexity in case of mesh network is better than all other schemes and is equal to 1, while sparse, arbitrary and proposed solution is nlgn and that of ring is n.

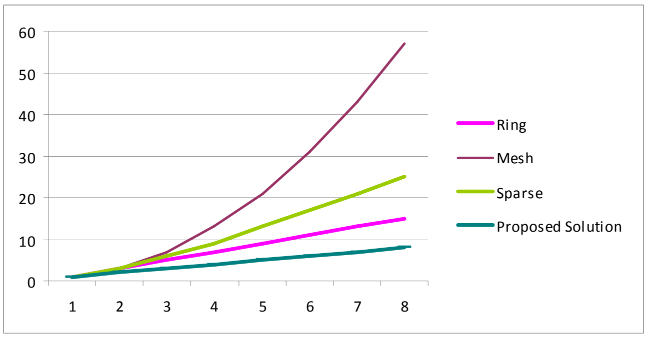

Fig. 5 Number of message passing

Figure 5 examines the number of message passing during each synchronization process for ring, sparse, mesh and proposed solution.

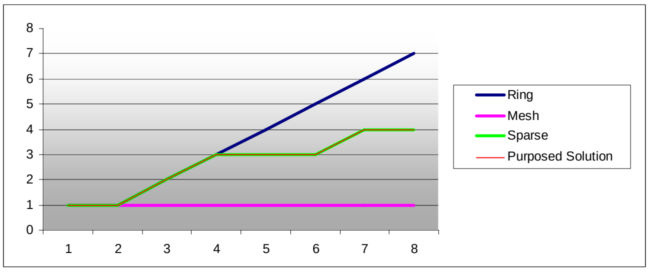

Fig. 6 Hop Count Complexity

Figure 6 examines the maximum hop count during each synchronization process for ring, sparse, mesh and proposed solution.

VII. Conclusion and Future Work

It is evident from simulation that in case of proposed solution the number of message passing for synchronization is equal to the number of sensor nodes in a sensor network, while traditional synchronization algorithms number of message passing is proportional to the number of edges between the sensor nodes,

So as far as message passing rate is concerned, our scheme offers the least number of messages to obtain the goal. Where as the complexity of hope is better than Ring network and equal to sparse network but higher than mesh.

In future we planned to enhance the proposed scheme to distance based synchronization.

References

[1] Bheemidi, D.R.; Sridhar, N "A Wrapper-Based Approach to Sustained Time Synchronization in Wireless Sensor Networks" Computer Communications and Networks, 2008. ICCCN '08. Proceedings of 17th International Conference on 3-7 Aug. 2008 Page(s):1 - 6 Digital Object Identifier 10.1109/ICCCN.2008.ECP.55

[2] J. Elson and D. Estrin, "Time synchronization for wireless sensor networks," in Proc. of the 2001 International Parallel and Distributed Processing Symposium (IPDPS), Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, Apr. 2001.

[3] Sichitiu, M.L.; Veerarittiphan, C., "simple, accurate time synchronization for wireless sensor," Proc. IEEE Intl. Conf. on Wireless Communications and Networking, 2003 IEEE (ICASSP), vol. 4, pp. 2033–2036, 2001.

[4] M. Lemmon, J. Ganguly, and L. Xia, "Model-based clock synchronization in networks with drifting clocks," in Proc. Of the 2000 Pacific Rim International symposium on Dependable Computing, Los Angeles, CA, Dec. 2000, pp. 177–185.

[5] K. Romer, "Time synchronization in ad hoc networks," in Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking and computing. 2001, pp. 173–182, ACM Press.

[6] Roberto Solis, V.S Borkar, P.R. Kumar, "A New Distributed Time Synchronization Protocol for Multihop Wireless Networks" Decision and Control, 2006 45th IEEE Conference on 13-15 Dec. 2006 Page(s):2734 - 2739

[7] B. Vukojevic, N. Goel, K. Kalaichevan, A. Nayak, I. Stojmenovic,. "Power-aware Depth First Search based georouting in ad-hoc and sensor wireless networks

[8] Mobile Wireless Communications Networks, 2007 9th IFIP International Conference on

[9] 19-21 Sept. 2007 Page(s):141 – 145

[10] C.-Y. Chong and S. Kumar, "Sensor networks: evolution, opportunities, and challenges," Proc. IEEE, vol. 91, no. 8, pp. 1247–1256, 2003.

[11] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), vol. 4, pp. 2033–2036, 2001.

[12] A. Ephremides and B. Hajek, "Information theory and communication networks: an unconsummated union," IEEE Trans. on Inform. Theory, vol. 44, no. 6, pp. 2416–2434, 1998.