Maintenance of Time Synchronization in Wireless Adhoc Networks in Case of Node Destruction

By Clock Sampling Mutual Network Synchronization

Swapna. P. S
M.Tech in Communication Engineering
Federal Institute of Science and Technology (FISAT)
Ernakulam, India

Bejoy Varghese
Dept. of Electronics & Communication
Federal Institute of Science and Technology (FISAT)
Ernakulam, India

***Abstract*—Software defined radios consists of nodes which are geographically separated radio terminal units that altogether form a TDMA communication network. These nodes exchange their timing information in the form of timestamps. Synchronization is achieved between the nodes from these timestamps. The main objective of this paper is to attain time synchronization between the nodes in the network by a Mutual Synchronization Technique named as Clock Sampling Mutual Network Synchronization Algorithm and also to maintain synchronization between the existing nodes when certain nodes gets destroyed from the network. This paper includes a modification on the Clock Sampling Mutual Network Synchronization to maintain synchronization in the case of destruction of nodes. Attaining time synchronization in case of node destruction finds its application in military communication enabling the existing nodes to communicate among them. The CSMNS algorithm to attain synchronization among the nodes and also to maintain it when certain nodes gets destroyed is simulated in MATLAB and its performance showing the error convergence towards zero is evaluated. Although an error in the range of microseconds arises in case of node destruction, with the algorithm the network attains synchronization in lesser number of iterations.**

***Keywords*— Wireless Adhoc Networks, TDMA, timestamp, Network Time Synchronization, CSMNS, JTRS.**

I. INTRODUCTION

The next generation radio systems used in military applications is based on software defined radios (SDR) which enables one radio to communicate with several kinds of radios, independent of the type of radios. Earlier soldiers were able to communicate with those who had same kind of radios. This problem of interoperability is solved with joint tactical radio systems (JTRS). Software defined radios consist of multiple software modules that allows the implementation of standards in the same radio system. Modulation, frequency ranges, bandwidth and other security functions is controlled by various kinds of software present in SDR. Software defined radio works as an ad hoc wireless TDMA communication, which comprises of geographically separated radio terminal units (or nodes) that can communicate among themselves over a wireless transmission medium. Every node is a collection of processors, memory, I/O units that coordinate among themselves in order to carry out a specific task. Nodes in the TDMA network make use of crystal as its clock. Deviation in clock timings of these nodes needs to be corrected for which synchronization is essential.

Mutual Synchronization technique [1] is a decentralized approach [2] where the node achieves synchronization by reducing its own timing error with respect to the weighted average of other clocks present in the network. Network Time Synchronization aligns time processes of different clocks in the network. First, wireless links of communication is established between the clocks comprising the network. Nodes generate a variable time process T that takes progressive values as time evolves. At a given time, the variable T in every node of the network must show the same value within some error tolerance. There can be variations in the oscillator's frequency due to which the clock may drift. Usually, for generating the time process we make use of crystal oscillators which is connected directly to the board of the processor. Imperfection in the crystal as well as temperature changes on various nodes The CSMNS algorithm runs throughout the network such that for any Ti and Tj with i ≠ j, $|Ti(t) - Tj(t)| \leq e$ where e is as close to zero and Ti denotes the time process of the ith clock. The main objective of this paper is to attain synchronization between the nodes in the network using CSMNS algorithm and also to modify it in order to maintain synchronization when certain nodes gets destroyed from the network, enabling communication between the existing nodes. In case of node destruction, communication should be maintained between the remaining nodes. So, it should be ensured that the time synchronization should be maintained between the existing nodes. For this purpose, a correction factor (named s) is passed on to the algorithm in case of node destruction. Even though an error in the range of microseconds arises when some nodes get destructed, the algorithm manages to maintain synchronization with lesser number of iterations. The algorithm does not require much energy to run and attain synchronization in changing radio propagation conditions which is an advantage. Also, the algorithm does not require any direct physical control of the clocks when the clock is multiplied with the correction factor.

The Network Time Synchronization Technique used in this work makes use of physical clocks to define the time process generated in all nodes of the network. The use of physical clock in the network does not imply interest in obtaining precise measure of time in terms of Coordinated Universal Time (UTC). For our purpose it is sufficient to obtain time synchronization within some error accuracy instead of the absolute value shown by the time processes.

II. PREVIOUS WORKS

Gersho and Karafin [3] proposed a scheme in which the system mutually synchronizes the phases of geographically separated oscillators that had already established links among each other. Each node in the network sends a train of periodic pulses that represent its timing process. The combined received signal at every node is averaged and fed in feedback in order to adjust the local oscillator's frequency. This method has a disadvantage that it demands for substantial bandwidth and energy resources when used over wireless systems.

Next, Hu and Servetto [4] came up with a new scheme in which the node located at centre of mass of the network transmits pulses that is circulated throughout the entire network. Each node compares the phases of their pulses and then correct their clocks to the time of the node located at the centre of mass of the network. But, it was not mentioned properly regarding the method of selection of node at the centre of mass of the network especially when the nodes move. Hence this method is also not applicable for wireless mobile networks.

In paper [5] a mutual network synchronization approach was proposed inter vehicular wireless networks. In this method, a square law envelop detector detects the power and relative delay of the narrow pulses in order to achieve synchronization. The node then computes a power weighted average of the delay between its own pulses and the ones received and the next pulse transmission time is either advanced or delayed according to the computed value. This method also requires a separate channel for synchronization tasks and also no studies have been made on its performance evaluation till now.

Finally, Carlos H Rental and Thomas Kunz [6], [8] came up with a new method of network synchronization called Clock Sampling Mutual Network Synchronization (CSMNS) which differs from the previous works and has no direct physical control of the clocks. The time adjustment made for correction is discrete and multiplicative and the use of timestamps for exchanging the timing information among nodes reduces bandwidth and energy requirements when compared to pulse transmission.

CSMNS is implemented in a method which does not require accessing of internal working of the nodes present in the network. Two main factors help to achieve it. Primarily, the use of virtual time and secondly, exchange of timestamp packets. Accurate timing correction of the clock skew adjustments is carried out by the multiplicative factor. Also CSMNS's core control law is quite simple and differs from the methods in [3], [4] and [5]. Practically, there are several approaches to exchange timing information among the different clocks present in the network which are as follows:

- Burst position method
- Continuous correlation of timing signals and
- Clock sampling method

In the burst position method, each node periodically transmits its bursts. At every receiving node, the difference in positions of the incoming bursts with that of the local bursts is used to correct the local clock period of the clock according to a many to one mapping method. The mapping method is in the form of a weighted average of the errors. This method requires a large bandwidth and a dedicated channel to carry out the task which is its drawback.

In the second method of continuous correlation, each node continuously transmits a signal which is received by the receiving node. At the received node, the sequence is compared with the replica generated by the local clock and a sliding correlation is carried out in order to calculate the phase offset. Similar to the previous case a many to one mapping is used to calculate the correction term, which is use to adjust the local clock.

In the Clock sampling method, every node reads the time of its clock and transmits it to the neighboring nodes that altogether form a communication network. At the receiving node the timing errors are computed by taking the difference between the local clock and the clock of the neighboring node. CSMNS makes the time difference among the clocks either converges to zero or close to tolerable error value by a multiplicative process. This algorithm makes use of a correction factor, named 's' to make the clock difference converge to zero. The main advantage of CSMNS algorithm is its simplicity.

III. TIME SYNCHRONIZATION BY CLOCK SAMPLING MUTUAL NETWORK SYNCHRONIZATION (CSMNS)

A TDMA communication network which consists of geographically separated nodes is considered for carrying out the simulation. All nodes exchange their timing information in the form of timestamps after establishing links of communication among them. CSMNS is the only NTS approach which is proposed for wireless ad hoc networks. Here, the algorithm is presented for a case of single hop wireless network. First, we will see how time synchronization is attained among the nodes in the network and secondly, we will discuss how the algorithm is used to maintain synchronization when certain nodes get destroyed from the network enabling communication between the existing ones.

A. *Mathematical Model for a Clock*

Oscillators are used for generating the time processes. The output of the oscillator is in the form of pulses when generated on PIC32MZ processor. The rising edge and the falling edge of the pulses are detected and added up by the accumulator which forms the time process. Fig 1 shows the model for a clock.

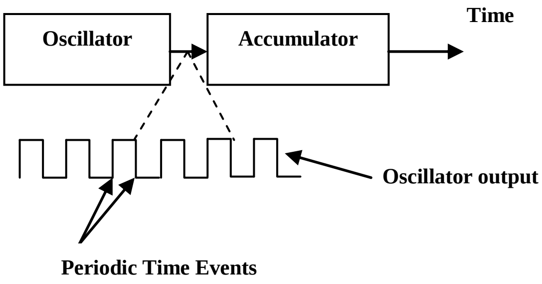

Fig 1. Clock model

B. *The CSMNS Algorithm*

The clock in the node that transmits the time stamp is considered as the transmitting clock whereas the clock that receives the timestamps is the receiving node. Time processes of the clocks are assumed to vary in a range of 25ppm. So, it is necessary to eliminate this error in the time processes of the clocks and hence attain time synchronization in order to ensure proper communication with the nodes.

The Clock Sampling Mutual Network Synchronization Algorithm (CSMNS) based on mutual network synchronization is described as follows:

- Every node consists of a real clock as well as a controlled clock. The controlled clock always reads from the real clock and adjusts the value read by a correction factor 's'. Synchronization information is always read from the controlled clock. The controlled clock is same as the real clock when s=1. Figure 2 shows the relationship between the controlled and the real clocks. As the network is based on TDMA, a node transmits its timestamps only in the specific timeslot allotted to it.
- After receiving timestamp from the specific node, the local node sets the timestamps of the controlled clock as well as the real clock. At the beginning of the algorithm, the value of 's' is set to1.

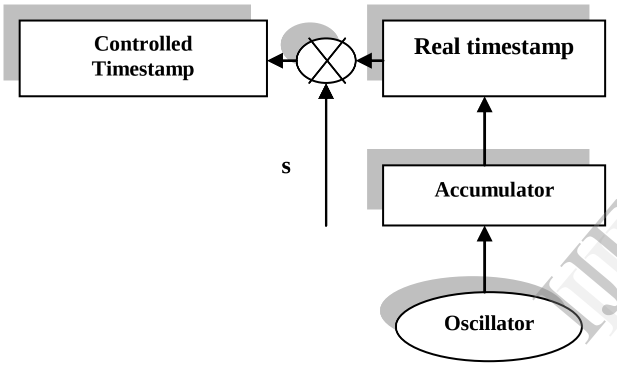

Fig 2. Controlled and real clocks in each node

- Suppose that ith node receives a timestamp successfully. The correction factor's' is computed from the difference of received timestamp from the local timestamp. The value of the correction factor's' is updated after the successful reception of every timestamp.

A more detailed analysis of the algorithm is described as follows:

The time process of the clock in every node is modeled by the following equation (1)

$$T(\tau) = \beta(\tau) + T(0) \qquad (1)$$

Where T is the time process of the clock, t is the real time, β is the skew of the clock with respect to real time and T(0) is the initial time of the clock.

Here, we assume a network of N nodes, each consisting of a clock that has different skew and initial time. This will result in a set of N equations of the form

$$T\iota(\tau) = \beta\iota(\tau) + T\iota(0) \quad ; \iota=\{1,2,3..\,.,N\} \qquad (2)$$

So, here our motive is to synchronize all the clocks in the network in such a way that after a time tc, |Ti(t>tc) – Tj(t>tc)| ≤ ΔT, where tc is the convergence time of the CSMNS algorithm and ΔT is the tolerable time error. The different βi in (2) make all the time processes Ti(t) to diverge as t→∞. The main goal CSMNS is to minimize the drifts of the time processes of various clocks present in the network. And this is achieved by multiplying the time process of the local clock by a correction factor si(t) which is computed by the following equation (3)

$$si(t) = si(t\text{-}1) + Kp[T'rxstp(t\text{-}1) – T'l(t\text{-}1)]/\, T'l(t\text{-}1) \qquad (3)$$

Where Kp is the proportional gain, T'rxstp is the timestamp of the node that successfully transmitted the timestamp and T'l is the timestamp of the local node i.e, the node which is receiving the timestamp.The correction factor is multiplied by the time process transforming Eqn(2) as

$$Ti'(t) = si(t)*\beta i(t) + si(t)*\, Ti(0) \quad \text{for all } i = \{1,2\ldots\ldots N\} \quad (4)$$

The concept of CSMNS algorithm can also be represented by the following block diagram as shown in Fig.3.

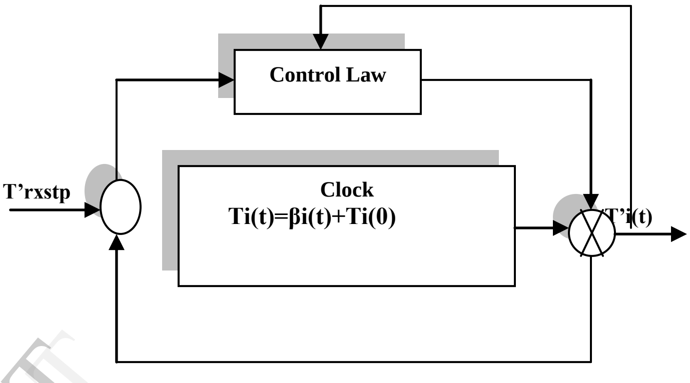

Fig 3. The CSMNS time offset scheme

An appropriate value of the proportional gain Kp will average the time processes of all the clocks in the network in order to achieve time synchronization among them.

Initial condition in Eqn(3) for si(0) is taken as 1. Ti(0) is uniformly distributed in the range [0, 100] and the skews βi is uniformly distributed in the range of ± 25ppm. The control law of the CSMNS algorithm is like a proportional controller that tracks the time processes of all the nodes present in the network.

C. Time Synchronization of nodes in the network by Clock Sampling Mutual Network Synchronization (CSMNS)

In this section, it is explained how synchronization is attained when all the nodes are present in the network using the basic Clock Sampling Mutual Network Synchronization Algorithm explained in the previous section. Suppose that the total number of nodes considered in the network which are to be synchronized is N. Out of the N nodes, one node wins the contention and this node is considered as the local node. The local node receives the timestamps of all the remaining N-1 nodes in specific timeslots, calculates the correction factor s and modifies the timestamp of the local node according to the equation (3). The process of multiplying the timestamp with the correction factor is carried out for several iterations until the time difference between various nodes become negligible or zero. The correction factor s is modified after receiving every timestamp.

D. Modification on CSMNS: Time Synchronization of existing nodes in the network when certain nodes gets destroyed from the previous network

In the base paper [7], "The Maintenance of TDMA Network Synchronization when the reference burst is vanished in K-JTDLS", it is mentioned that the clock difference information between the primary station's local clock and node clock before the disappearance of the reference burst is passed onto the control block which restores the variation rate of the sub frame by averaging the clock difference information after the reference burst is vanished. The control parameter which has been passed to the control lock is nothing but the clock difference information. Similarly, in this case also a control parameter needs to be passed on to the algorithm when comes to a case of destruction of some nodes from the whole network maintaining synchronization between the existing nodes.

The correction factor s is the control parameter here which is being passed. The last value of s computed before the destruction of the nodes is passed onto the algorithm when some nodes have been destroyed and with some modifications in the algorithm the network synchronization is attained with the remaining nodes. The various case that can exist in a network is as described as follows:

Suppose that the N nodes in the network are synchronized, out of which some nodes gets destroyed (say three nodes). Now the timing information (timestamp) from the three nodes is lost and the network has lost its state of synchronization. But, in order to maintain communication with the remaining nodes, synchronization must be achieved with the remaining nodes. This can be achieved as follows:

- Receive the timestamps in the same manner as that of the case of N nodes (in earlier case).
- Leave the timeslots of the destroyed nodes as vacant and let the winning node receive the timestamps of the existing ones as in the same manner as before.
- Compute the correction factor s according to the differences between the timestamp of the local node and received nodes. In this case the initial value of s is the last modified value of s when N nodes have achieved synchronization.
- Multiply the s factor with the local node and repeat the process until the difference between the timestamps is negligible or zero.

Now consider the case when only two nodes exist in the network i.e, the local node and the one which is transmitting the timestamps. It is considered that rest of the nodes is destroyed. Still communication is to be maintained with the existing two nodes.

- Previous value of s (when synchronization has been achieved in the previous case) is taken.
- Difference between the timestamps of the local and received nodes is computed and s factor is calculated. The timestamp is sent only in their respective slots and rest of the timeslots is left vacant.
- The local timestamp is modified by the calculated s factor and the process is repeated until the time difference between the nodes become negligible.

Although there will be some fluctuations from the synchronized state when some nodes are destroyed, the algorithm synchronizes the time processes of the remaining nodes after some iterations. This is relatively a simple process to attain synchronization between nodes in a network when some of them are destroyed.

IV. RESULTS AND DISCUSSION

The duration of a frame generated by a node is taken as 0.6seconds. Simulations were done in MATLAB and the following results were obtained. The TDMA system clock (sample clock) is taken as 20 MHz in the simulation.

The graph shown in Fig 4 shows the convergence of error values between time processes of two nodes with respect to the number of iterations after applying CSMNS algorithm, for different values of Kp. From the graph, it is clear that as the value of Kp increases, the time taken (in terms of number of iterations) for error value to become zero decreases. For Kp=0.5, the number of iterations required to eliminate the error is 3, whereas for Kp=0.1 it increases to more than 5.

Error value = received timestamp – local timestamp

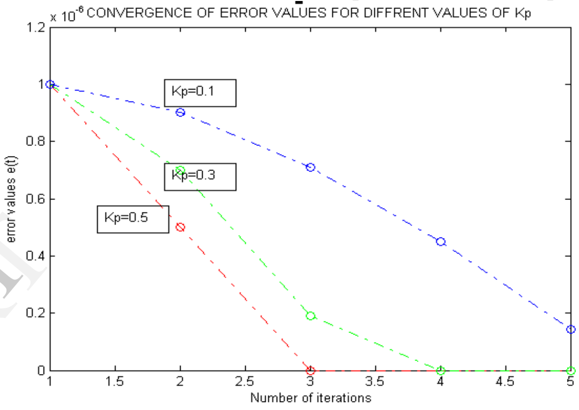

Fig. 4. Convergence in error values of two nodes after applying CSMNS time offset and frequency offset scheme

Fig 5 deals with error convergence when the total number of nodes in the network is 10. In the simulation, node 1 wins the contention and is considered as local node which receives timestamps from rest of the 9 nodes. The curve shows the error convergence towards zero of node 1 and hence four iterations are required to make the timestamp difference between first and other 9 nodes as zero and hence the network is said to be synchronized

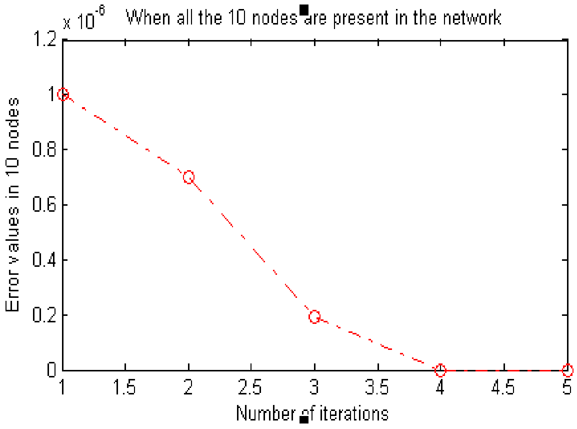

Fig 5. Error convergence when all the ten nodes are present in the network

After sometime, consider that 3rd, 4th and 5th nodes are destroyed from the network. The error gradually increases, but after applying the algorithm, it again decreases to zero in the next three iterations. Hence again the network is synchronized. This condition is shown in Fig 6.

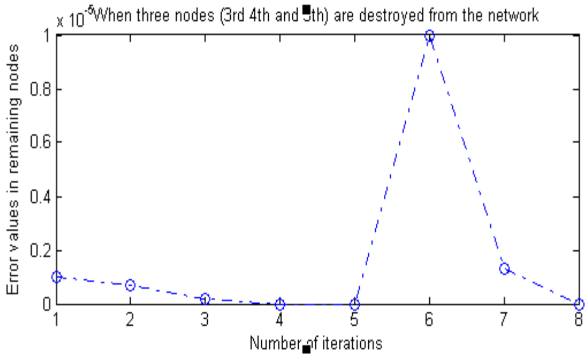

Fig 6. when the 3rd, 4th and 5th nodes are destroyed from the network of 10 nodes

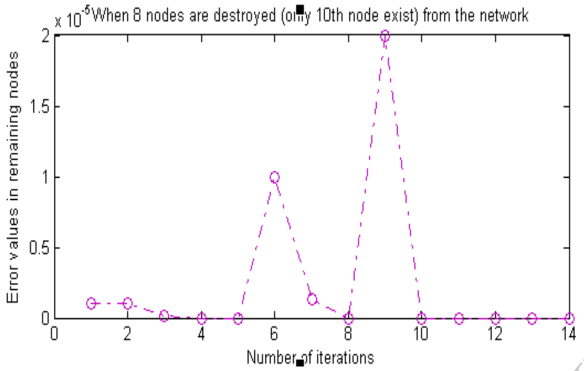

Fig 7. when only 10th node exists (nodes from 2 to 9 are destroyed) in the network

Now, let all the nodes (except 1st and 10th node) get destroyed. This case is shown in Fig 7. The error increases again and then converges to zero by making the network synchronized. The algorithm is performed for a value of Kp=0.3.

Note that the number of iterations required (time required) by the algorithm to make the error towards zero (i.e. to make the network synchronized) increases as the number of nodes participating in the network increases. But the error value increases as the number of nodes decreases.

V. CONCLUSION AND FUTURE WORK

To summarize, it has been shown that a node multiplies its time process by a multiplicative factor computed using the recursion in (4) and therefore time synchronization is attained between the nodes by making the difference between the time processes negligible and also to maintain it when certain nodes gets destroyed from the network enabling communication between the existing nodes. Any timestamp successfully received in every slot is used to update the multiplicative factor. CS-MNS nodes will synchronize with any value of time stamps. Also, the algorithm is simple and can synchronize between existing nodes in the network even though some of them are destroyed from the whole network. This finds application in military communication ensuring communication between the existing nodes when some nodes are destroyed. The most important advantage is that this method can even synchronize from any number (here taken as 10) of nodes to a minimum of two nodes in the network. As a future work, this algorithm can be implemented using PIC32MZ processor which has two clocks in it.

ACKNOWLEDGMENT

This work was made possible with support from Central Research Laboratory, Bharat Electronics Limited, Bangalore.

REFERENCES

[1] W.C Lindsey, F. Ghazvinian, W. Hagmann, and K. Dessouky, .Network synchronization,. Proceedings of the IEEE, vol. 73, No. 10, pp. 1445-1468, October 1985

[2] W.C. Lindsey and J.-H. Chen, “Mutual Clock Synchronization in Global Digital Communication Networks,” European Trans. Telecommunications, vol. 7, no. 1, pp. 25-37, Jan./Feb. 1996.

[3] A. Gersho and B.J. Karafin, “Mutual Synchronization of Geographically Separated Oscillators,” Bell Systems Technical J., vol. 45, pp. 1689-1904, Dec. 1966.

[4] A. Hu and S.D. Servetto, “Asymptotically Optimal Time Synchronization in Dense Sensor Networks,” Proc. Second ACM Int’l Workshop Sensor Networks and Applications (WSNA ’03), pp. 1-10, 2003.

[5] E. Sourour and M. Nakagawa, “Mutual Decentralized Synchronization of Intervehicle Communications,” IEEE Trans. Vehicular Technology, vol. 48, no. 6, pp. 2015-2027, Nov. 1999.

[6] Rentel and Kunz, “ A Mutual Network Synchronization Method for Wireless Adhoc and Sensor Networks”, IEEE transactions on Mobile Computing. Vol. 7. No.5, May 2008

[7] Jinwoo Han, Jae Pil, Lee Jang Dhongwoon , Oh Sangkyun ,Ilhyuk Oh “The Maintenance of TDMA Network Synchronization when Reference Burst is Vanished in K-JTDLS” The 2011 Military Communications Conference- Communications and Network Systems

[8] Carlos H.Rentel and Thomas Kunz,” Network Synchronization in Wireless Adhoc Networks”Technical Report SCE-04-08,July 2004.