2008 International Symposium on Nonlinear Theory and its Applications
NOLTA'08, Budapest, Hungary, September 7-10, 2008

Mode-Lock Eliminating Timing Synchronization Algorithm for Intervehicle Ad-hoc Networks

Kenta Shinohara† and Hisa-Aki Tanaka†

†Department of Electronic Engineering, The University of Electro-Communications (UEC)
1-5-1 Choufugaoka, Choufu-shi, Tokyo 182-8585 Japan
Email: shinohara@synchro2.ee.uec.ac.jp

Abstract—Timing synchronization is an important integrating component in mobile ad-hoc networks and also in sensor networks, and therefore, various timing synchronization algorithms have been proposed so far. Recently, Imai and Suzuki developed an efficient synchronization algorithm for vehicle-to-vehicle communications based on time division multiple access (TDMA) protocol. Despite of its efficiency in synchronization for vehicle-to-vehicle communications, their algorithm sometimes suffers from a certain undesired asynchronous pattern; so called mode-lock state. Although their algorithm takes some precaution to avoid this mode-lock state, in around more than 10% of instances this state still remains and synchronization cannot be obtained. Then, we clarify how and why this mode-lock state remains in their algorithm. Based on this insight, we obtain a new mode-lock eliminating, distributed algorithm that always leads to a perfect synchronization. From systematic, comparative simulations, it is observed that our proposed algorithm always eliminates mode-lock states, and realizes even faster synchronization, compared with the algorithm by Imai and Suzuki.

1. Background of this study

Applications of mobile ad-hoc networks are actively developed for vehicle-to-vehicle communications recently. For instance, in Toyota systematic researches are carried out, which cover the timing synchronization[1], MAC protocol[2], and even urban traffic simulations for intervehicle ad-hoc networks. Recently, Imai and Suzuki developed an efficient synchronization algorithm for vehicle-to-vehicle communications based on time division multiple access (TDMA) protocol[1]. Their algorithm includes some new devices on synchronizing two (or even multiple) groups of vehicles and its effectiveness is verified for certain environments in simulation. However, as they point out in [1], their algorithm sometimes suffers from a certain undesired asynchronous pattern; so called mode-lock state. Although their algorithm takes some precaution to avoid this mode-lock state, in around more than 10% of instances this state still remains and synchronization cannot be obtained in [1]. To guarantee a perfect synchronization, this study first considers how and why this mode-lock state remains in their algorithm. Based on this insight, a new mode-lock eliminating, distributed algorithm is realized that always leads to a perfect synchronization. From systematic, comparative simulations, we observe the proposed algorithm always eliminates mode-lock states, and realizes even faster synchronization, compared with the algorithm by Imai and Suzuki.

This report is organized as follows. In Section II, we explain the distributed timing synchronization algorithm proposed by Imai and Suzuki. In Section III, we consider and clarify how and why the mode-lock state remains in their algorithm. Then, we propose a simple mode-lock eliminating, synchronization algorithm. In Section IV, we verify the performance of both algorithms by systematic simulations. Finally, conclusions and discussions are given in Section V.

2. Basic design of distributed, timing synchronization algorithm by Imai and Suzuki

For distributed timing synchronization in vehicle-to-vehicle communications, there have been a few studies reported so far [1], [3], [4]. These synchronization methods are basically based on a kind of ‘averaged timing ’ over mobile nodes directly communicating each other. This averaged timing is characteristic of distributed, intervehicle communications, which enables each mobile node update its timing to the average of (a set of past, received) timing messages from neighbouring nodes. By this synchronization method using averaged timings, groups of vehicles ‘smoothly’ synchronize their timings each other, and therefore this method is expected to be suitable for TDMA protocol in intervehicle communications, as opposed to other synchronization methods in IEEE 802.11 and so on.

The synchronization method by Imai and Suzuki [1] is equivalent to the previous methods [3], [4] in principle, as these are based on averaged timing. However, this method includes two new additional devices as follows, and this method can be advantageous over previous methods when applied to larger, distributed intervehicle communication networks;
(i) this method takes into account of the synchronization process between two (or more) groups of vehicles, and introducing a certain acceleration effect for synchronization in each node, and also
(ii) this method takes some precaution to avoid a certain

undesired asynchronous state; so called mode-lock state.

For the above point (i), we will provide a detailed description later in our presentation, and as for the point (ii), explanation and analysis of it will be given in the next section.

3. A mode-lock eliminating timing synchronization algorithm

As pointed out in [1], undesired asynchronous states; so called mode-lock states emerge in Imai and Suzuki's synchronization method. Although this mode-lock has some variations in its structure, the essential point is schematically illustrated as in Fig. 1. In this figure, each node is placed on the regular 2D lattice (just for simplicity of illustration) and 'core' appears in the center of it (also just for simplicity). Since we have assumed a TDMA protocol (shown in Fig. 2) as a basis of this study, each node has the same period of packet transmission (; frame length in Fig. 2), and neighbouring nodes (node A, B, C, and D) should have distributed timings of transmission each other. This TDMA protocol is based on the one and the essential assumption; each node is time synchronized each other within a precision of guard interval. However, in the mode-lock state of Fig. 1, this assumption is violated because each node has an equally distributed timing of the clock (NOT the timing of packet transmission in Fig. 2) just like a rotating wave around the core. This rotating, mode-lock state is notorious in several branches of engineering; VLSI clocking [5], millimeter wave generation [6], heart diseases [7], and so on.

Imai and Suzuki are aware of this undesired state and include the following precaution in their method;
(i) each node determines if it is inside or near a possible core of mode-lock state by checking a threshold of the distribution of (a set of past) received timing messages from neighbouring nodes, once in a frame period, and then
(ii) if a node is aware of being inside or near the core, this node immediately updates its clock timing to the most recently received timing from neighbouring nodes.

However, intuitively speaking, their precaution is nothing but a king of random perturbation to the core of mode-lock state, and this does not guarantee to remove the mode-lock itself, since this state is known to be quite stable and robust, as long as the random perturbation is limited to nodes around the core[6].

Instead of the above random perturbing approach, we have investigated other effective mode-lock 'eliminating' synchronization methods. Below, we propose a simple, but quite effective synchronization method that destabilizes the mode-lock itself. The proposed method is basically on the same line of previous methods [1], [3], [4], as it is based on the averaged timing from neighbouring nodes. However, as opposed to the previous methods, this method intentionally eliminates the mode-lock and realizes a perfect synchronization in a distributed manner as follows;

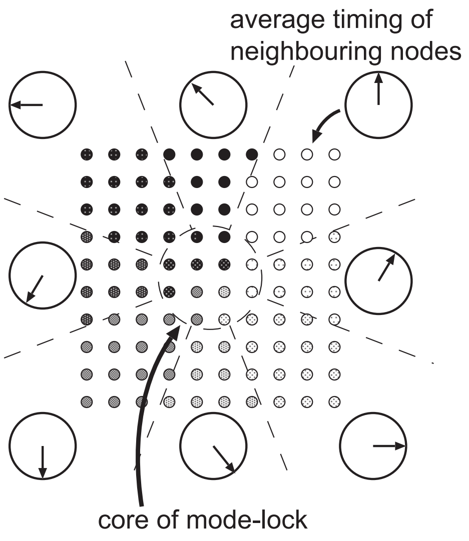

Figure 1: Basic structure of mode-lock state

(i) each node determines if it is inside or near a possible core of mode-lock state by checking a threshold of the distribution of timing messages from neighbouring nodes, and at this stage this method is totally the same to the method by Imai and Suzuki. However, contrary to their method,
(ii) if a node is aware of being inside or near the core, this node temporally quits sending its erroneous timing messages and only receives timing messages from other nodes.

This algorithm is quite simple in itself. However, when all nodes start synchronizing each other by this algorithm in the network, the effect of it drastically appears because the mode-lock state gradually loses its stability and is eventually totally wiped out. A detailed analysis and systematic comparison of both methods will be presented in the next section.

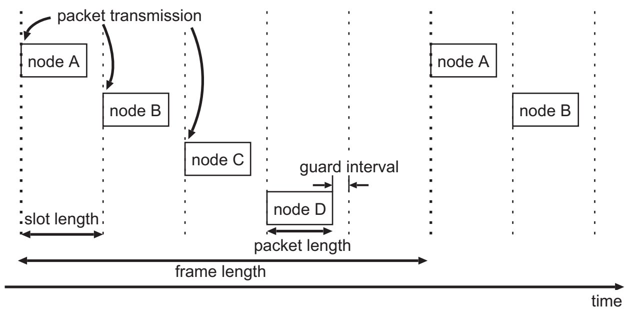

Figure 2: TDMA protocol in this study

4. Performance analysis for both algorithm

To evaluate the performance of both methods, comparative simulations are carried out systematically. The simulation setup in this study faithfully follows the setup in [1], which is described as follows.
(i) Each node periodically sends a timing message to neighbouring nodes in the beginning of its assigned frame (Fig.

2). In this simulation, one frame length is set to 0.1 [s].
(ii) Each node has a small frequency mismatch in its timing clock, which ranges in ±0.01% accuracy.
(iii) Each node is uniformly placed on a 2D lattice, and each node can communicate with neighbouring nodes in a (1) two-hop distance or (2) 2.5-hop distance. In this simulation, we consider a relatively short time scale of events in widely distributed, urban traffic environments. Therefore, mobility of each node can be practically ignored.
(iv) Following the setup in [1], the total number of nodes is set to 1024. Each node with initial random timing of frame is introduced to the 2D network, one by one for 100-frame time span in the beginning of the simulation.
(v) As a first approximation, a collision between timing message packets is ignored. (However, this will be taken into accounts, later in a more detailed simulations.)

4.1. Eliminating process of mode-lock states

First, to correctly visualize and measure the synchronization process, it is convenient to introduce a ‘ unit circle ’ as shown in Fig. 3. On this unit circle, the timing of each node is identified to a certain point of the [0, 2π] span. Namely, the periodic event in each node (in Fig. 2) is (uniquely) mapped to a ‘phase point’ on the unit circle. Then, synchronization process for each method in this study can be clearly visualized by the temporal variation of ‘phase point’ on the unit circle, and a fair comparison among different synchronization methods become possible by this coordination.

A typical example of mode-lock eliminating process in the proposed method is visualized as in Fig. 3. Figure 3(a) shows the initial mode-lock state, where seven ‘core’ nodes (○) are detected by this algorithm and other nodes (×) exhibit uniformly distributed timings, showing a stable rotating wave around the core. However, as this synchronization method operates, a ‘cut’ appears in the initial uniform distribution of timings as shown in Fig. 3(b). And once this cut appears, it grows rapidly (Fig. 3(c) and 3(d)), and finally the mode-lock is totally eliminated and perfect synchronization is stably realized. What is observed in the above example (and all other numerous data) is summarized as follows.
(i) In the presented method, when the core is detected it is virtually removed from the network for certain periods of frame, and the mode-lock on the remaining nodes temporally decreases its stability and initial uniform distribution of timing.
(ii) Then, once a ‘cut’ (in Fig. 3(b)) appears due to temporal fluctuations, nodes around this cut are included to the core by this algorithm. This again decreases a stability of the mode-lock, and therefore, elimination of the mode-lock is rapidly processed later on.

Contrary to our method, the original synchronization method by Imai and Suzaki does not have any such destabilizing function. Instead, their method includes a kind of randomly perturbing effect to the core of mode-lock, but this does not guarantee the elimination of mode-lock, as mentioned in Section 3.

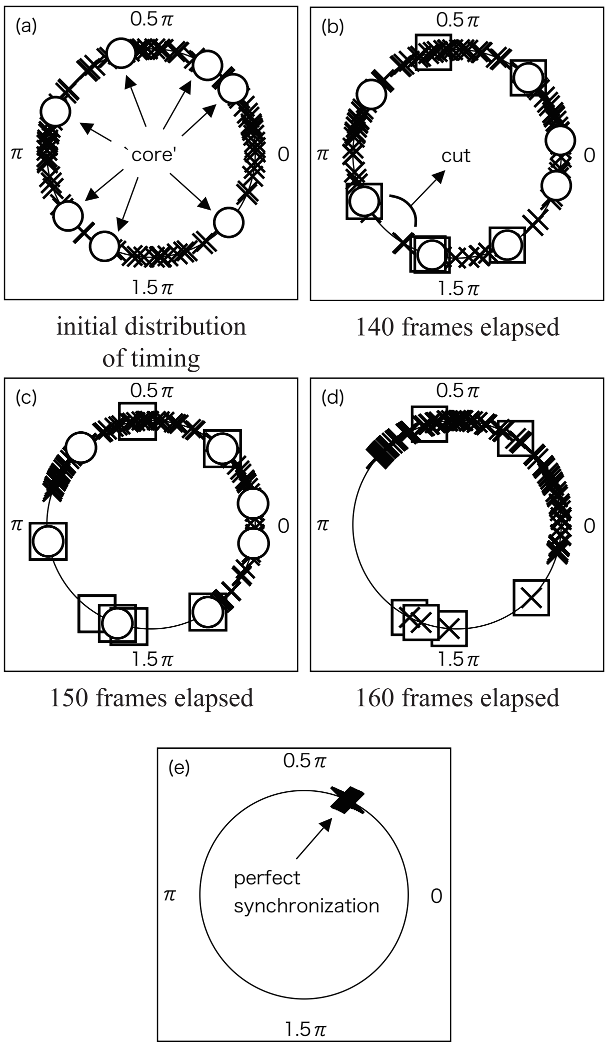

Figure 3: Eliminating process of mode-lock state

4.2. Synchronization ability for both algorithms

Thus far, we have reached to an insight of essential difference for both methods, which suggests that the presented method provides better synchronization ability, compared with the original method by Imai and Suzuki.

As mentioned in the previous subsection, to measure the degree of synchronization for all nodes, it is convenient to introduce the phase of timing, by which each timing of the node is identified to a certain phase between the [0, 2π]-span.

Using this coordination, degree of synchronization σ can be clearly defined as

$$\sigma(t) = \frac{1}{N}\left|\sum_{i=1}^{N} e^{j\phi_i(t)}\right|, \tag{1}$$

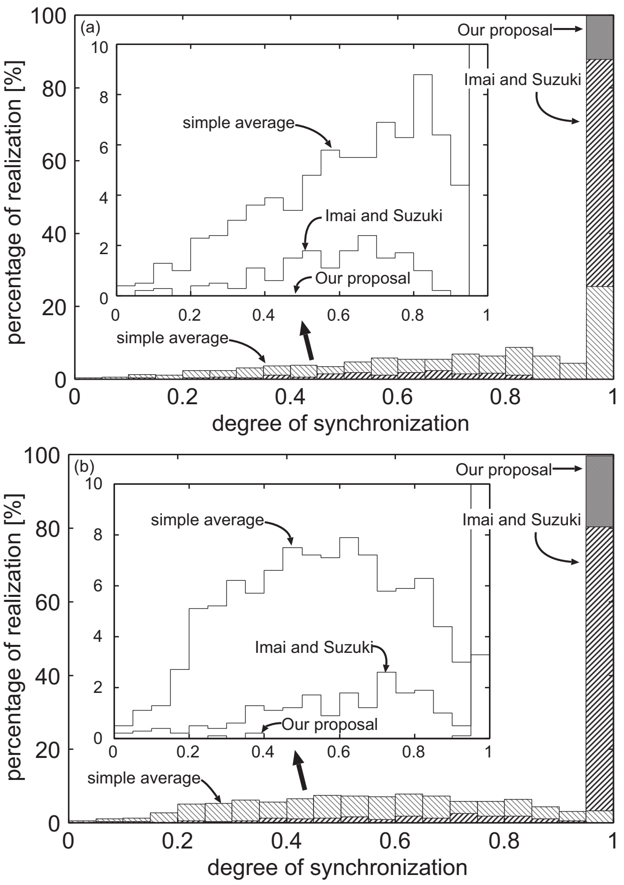

Figure 4: synchronization ability for both algorithms

where $\phi_i(t)$ represents the phase of the i-th node at time t and N is the total number of nodes. By this definition, σ becomes 1 when all nodes are perfectly synchronized, and becomes 0 when their phases (; timings) are uniformly distributed. It is noted that this σ takes a value between 0 and 1, and as long as the mode-lock remains σ always become less than 1.

Below, systematic simulations are carried out to compare the performance for both algorithms, by using this σ, Figure 4(a) and 4(b) show the result of 1,000 random trials for both methods for cases of 2.5-hop transmission range and 2-hop transmission range, respectively. In each trial, simulation is carried out for 500-frame periods, and the percentage of realization for final degree of synchronization σ is summed up from 1,000 trials. Besides the comparison between Imai and Suzuki's and our methods, 'simple average' method is also considered in this simulation. This simple average, method implies a synchronization method without any precautions to mode-lock. Then, compared with this simple average method, the effect of mode-lock elimination becomes transparent for other two methods.

In Fig. 4(a), the presented method realizes a 100% synchronization for 1,000 trials, while in Imai and Suzuki's method more than 20% instances remains unsynchronized and this shows the mode-lock is not eliminated for these cases. Also, in Fig. 4(b), nearly perfect synchronization is realized by the presented method. It should be noted that, for a few instances out of 1,000 trials, mode-lock is not completely eliminated by this method within 500-frame periods. However, we have verified these mode-locks can be eliminated by an additional precaution or by simply taking more time.

5. Conclusions

We have presented a simple, but quite effective synchronization method for intervehicle ad-hoc networks. This method is based on an insight from nonlinear dynamics of mode-lock states, and its effectiveness is supported by a series of comparative simulations with other synchronization methods. Further studies are currently ongoing as;
(i) more rigorous foundations for effectiveness of the presented method,
(ii) more detailed simulation including packet collisions and other node placements, and
(iii) simplification of this method for wireless sensor networks.

Some of them will be presented in our talk at NOLTA.

References

[1] J. Imai and N. Suzuki, "Study on a Self-Adaptive Timing Synchronization : A Method to Avoid Local Optimizations," IEICE technical report, USN2007-13, pp. 67 – 71, May 2007.

[2] S. Makido, N. Suzuki, T. Harada, and J. Muramatsu, "Decentralized TDMA Protocol for Real-time Vehicle-to-Vehicle Communications," Transactions of Information Processing Society of japan, Vol. 48, No. 7, pp. 2257 – 2266, Jul 2007.

[3] Y. Akaiwa, H. Andoh, and T. Kohama, "Autonomous Decentralized Inter-Base-Station Synchronization for TDMA Microcellular Systems," *Vehicular Technology Conference, 1991. Gateway to the Future Technology in Motion., 41st IEEE*, pp. 257 – 262, May 1991.

[4] E. Sourour and M. Nakagawa, "Mutual Decentralized Synchronization for Intervehicle Communications," *IEEE Trans. on Vehicular Technology*, Vol. 48, No. 6, pp. 2015 – 2027, Nov 1999.

[5] V. Gutnik and A. Chandrakasan, "Active GHz Clock Network using Distributed PLLs," *ISSCC Dig. Tech. Papers*, pp. 174–175, Feb. 2000.

[6] H. Tanaka and A. Hasegawa, "Modelock avoiding synchronisation method," *IEE Electronics Letters*, vol. 38, no. 4, pp. 186 – 187, 2002.

[7] N. Bursac, F. Aguel, and L. Tung, "Multiarm spirals in a two-dimensional cardiac substrate," *Proceedings of National Academy of Science*, Vol. 101(43), pp. 15530–15534, 2004.