2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

Mutual synchronization over groups of vehicles in intervehicle ad-hoc networks

Kazu-Aki Akashi†, 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: k-akashi@synchro2.ee.uec.ac.jp

Abstract—In this study, we consider timing synchronization methods over groups of vehicles, which is important in intervehicle communications [1], [3]. Here, we clarify whether a proper synchronization is always realized and this is stably maintained, analytically and numerically, when the Imai and Suzuki's synchronization algorithm[1] is applied. As a result, it is shown that the proper synchronization is stably obtained for the case of two groups of vehicles, but it is not always obtained for three groups of vehicles. Also, as a byproduct, a refined synchronization algorithm is constructed, realizing better synchronization ability.

1. Background of this study

Applications of mobile ad-hoc networks are now actively developed for vehicle-to-vehicle communications. For instance, systematic researches have been carried out to realize decentralized TDMA (Time Division Multiple Access) intervehicle communications, which cover the timing synchronization [1], and MAC protocol [2]. Recently, Imai and Suzuki developed an efficient timing synchronization algorithm [1] for vehicle-to-vehicle communications based on a TDMA protocol [2]. Their synchronization algorithm includes some new devices on efficiently synchronizing all timers in two (or even multiple) groups of vehicles and its effectiveness is verified for certain environments in their simulations. However, as they point out in [1], their algorithm sometimes suffers from a certain undesired synchronous 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 [1]. In this study, we focus on more realistic situations where multiple groups of vehicles mutually synchronize all their timers. This situation is practically important in intervehicle communications, and has been studied so far [3]. However, even in such rather simple situations, it is not obvious if a proper synchronization is always realized and this synchronization is stably maintained. Here, we clarify this issue analytically and numerically, when the Imai and Suzuki's synchronization algorithm is applied. We show the proper synchronization is stably obtained for the case of two groups of vehicles, but it is not always obtained for three groups of vehicles. Also, as a byproduct, a refined synchronization algorithm is constructed, realizing better synchronization ability.

This paper is organized as follows. In Section 2, we briefly review the conventional distributed timing synchronization methods proposed for intervehicle communications. In Section 3, we formulate the problem where multiple groups of vehicles synchronize each other, and we clarify when the mode-lock state stably remains in their algorithm. Then, we propose a simple, refined synchronization algorithm to destabilize the mode-lock state. In Section 4, we analyze the stability of synchronous patterns for the Imai and Suzuki's synchronization algorithm. Finally, conclusions and discussions are given in Section 5.

2. Conventional distributed, timing synchronization methods and new synchronization algorithms by Imai and Suzuki

For distributed timing synchronization methods in vehicle-to-vehicle communications, there have been a few studies reported so far [1], [3], [4]. These synchronization methods are basically based on the ‘averaged timing’ over mobile nodes (; vehicles in this study) directly communicating each other, which is literally defined by the average of all local times (timings) received from communicating nodes; namely the average of a set of the most recently received timing messages from neighbouring nodes. Then, by using this averaged timings, it is expected that groups of vehicles gradually and smoothly synchronize their timings each other, and therefore this method is assumed to be suitable for the TDMA protocol in intervehicle communications [2]. This feature shows a contrast to other synchronization methods such as timing synchronization function (TSF) in IEEE 802.11 MAC, which uses only the fastest timing from communicating nodes.

The synchronization method by Imai and Suzuki [1] is equivalent to the previous methods [3], [4] in principle, as these are based on the 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, because
(i) this method takes into account of the synchronization process between two (or more) groups of vehicles, and introducing a certain acceleration into the synchronization procedure in each node, and also because
(ii) this method takes some precaution to avoid a certain undesired synchronous state; so called a 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. Synchronization of multiple groups of vehicles

In this section, we start by formulating the problem regarding how multiple groups of vehicles mutually synchronize each vehicle's timer. Then, we carry out systematic numerical simulations for three typical and the most commonly observed instances of group synchronization, to clarify if the proper synchronization is always realized or not, and to estimate how quickly this synchronization is attained.

3.1. Problem formulation

Here, we consider the three typical and the most commonly observed instances of group synchronization as shown in Fig. 1. Figure 1 (a) and (b) respectively show the case of two groups and three groups of vehicles driving to the same direction, within a line of sight (LOS). Also, Fig. 1 (c) shows three groups of vehicles meeting at the intersection.

In each case, the time scales of synchronization is much shorter than that of changing positions of vehicles during group synchronization (which is explained later). Then, it is natural to assume the area of mutual communications of groups arises between the terminal vehicles in each groups, as shown in Fig. 1. Also, since each group is formed before they meet each other, it is reasonable to assume synchronization is attained in each group already.

Then, what we want to clarify is summarized as follows.
(i) Are all the groups always led to the proper (; perfect) synchronization for any initial conditions of timers in each vehicles ?
(ii) How quickly do all the groups attain the synchronization ? In other words, how long do we have to wait until the synchronization is obtained ?

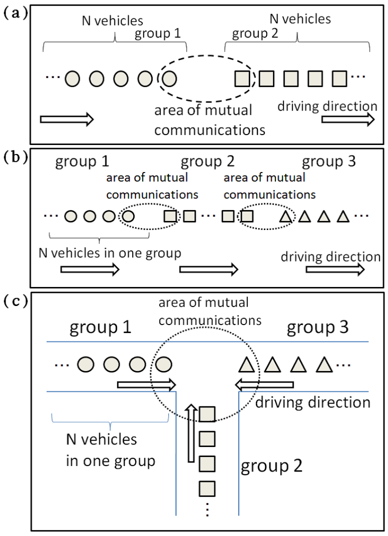


Figure 1: Mutually communicating groups of vehicles

3.2. Systematic simulations for synchronization process

To evaluate the performance of the Imai and Suzuki's algorithm, comparative simulations are carried out systematically. Details of the simulation setup are summarized as follows.
(i) As mentioned in 3.1, we assume vehicles in each group of Fig. 1 have mutually synchronized their timings at the beginning of the simulation. The initial timings of each group in Fig. 1(a), (b) and (c) are respectively defined later in this subsection.
(ii) Each vehicle periodically transmits a timing message to neighboring vehicles at the beginning of its assigned slot once per a frame as shown in Fig. 2. In this simulation, one frame length is set to 0.1 [s]. Collisions or lost of these messages are neglected as a first approximation in this simulation.
(iii) We assume each vehicle is uniformly placed in its group, for simplicity. In each group, each vehicle can communicate with neighboring vehicles within the R-hop distance (R is set to 8 in this simulation for instance). Also, only the terminal vehicles (circled in Fig. 1) can communicate with terminal vehicles in other groups. Here, we consider a relatively short time scale of events (up to a second) in widely distributed, urban traffic environments. Therefore, mobility of each vehicle can be ignored as a first approximation in these simulations, since its mobility is relatively small with respect to the range covered by the groups of vehicles.
(iv) We assume N vehicles in each group, and N is set to 40, for instance.
(v) In the decentralized TDMA protocol[1], each vehicle is assigned a fixed slot for periodic packet transmissions, and the mutual synchronization between vehicles is required within the precision of the guard interval in Fig. 2. Then we regard every two vehicles being synchronized within the range of the guard interval as properly synchronized, in this study. The guard interval is set to ± 8% of the slot length for instance, in this simulation.

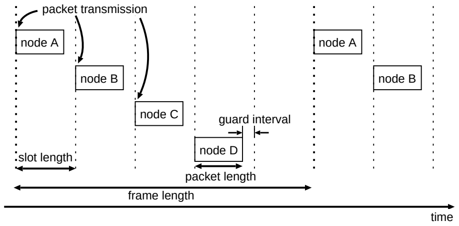
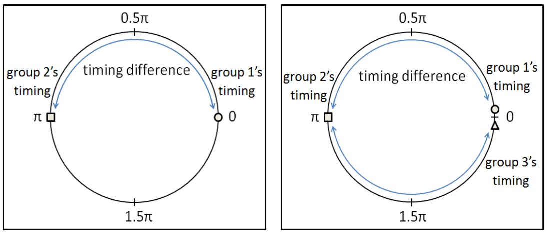


Figure 2: Time course of TDMA protocol in this study

Then, we analyze the performance of the Imai and Suzuki's algorithm for the case of two groups and three groups within LOS as shown in Fig. 1(a) and (b). Here, to correctly visualize and measure the synchronization process, we introduce a 'unit circle' as shown in Fig. 3. On this unit circle, the timing of each vehicle is identified to a certain point of the [0, 2π] span. Namely, the periodic event in each vehicle (in Fig. 2) is (uniquely) mapped to a 'phase point' on the unit circle. Then, synchronization process for each algorithm 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 algorithms become possible with this coordination. In the simulation of two groups, as the worst case, the initial timing difference between two groups is set to the half of the slot length as shown Fig. 3(a), which is the maximum of possible timing difference. Figure 4 shows the timing distribution right after synchronization is realized, for two groups within LOS. The 'simple averaged' algorithm in Fig. 4(a) implies the Imai and Suzuki's algorithm without acceleration for synchronization between groups (mentioned in Section 2 (i)).

Similarly to the case of two groups, we analyzed the case of three groups within LOS. The initial timing distribution is given, as shown in Fig. 3(b). This setup is considered as one of the worst cases where each group has the maximal timing difference with each other. Figure 5 shows the timing distribution right after the synchronization is realized, for three groups within LOS. As shown in Fig. 4 and 5, proper synchronization is verified both for the case of two and three groups within LOS.



(a) Case of two groups (b) Case of three groups

Figure 3: Initial timing distribution for simulations

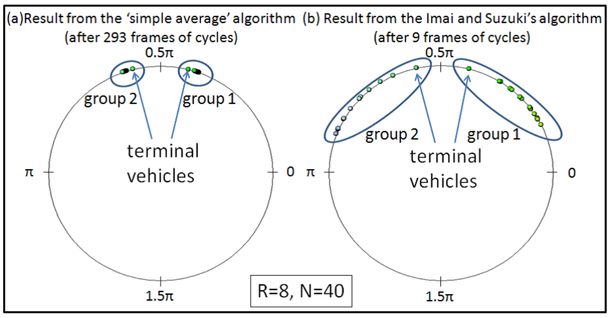


Figure 4: Timing distribution right after the synchronization is realized, for two groups within LOS

As opposed to the case of vehicles within LOS (Fig. 1(a) & (b)), we consider here the case of three groups meeting at the intersection as shown in Fig. 1(c). Figure 6 shows the timing distribution after 1000 frames of cycles for this case, starting from the initial timings shown in Fig. 3(b). When the 'simple averaged' algorithm is applied, although vehicles in each group are completely synchronized each other, the timings of three groups are evenly spaced on [0, 2π] and this state is stabilized as in Fig. 6(a). Even if the Imai and Suzuki's algorithm is applied, this undesired synchronous pattern is maintained throughout the simulation as shown in Fig. 6(b), and complete synchronization is never obtained.

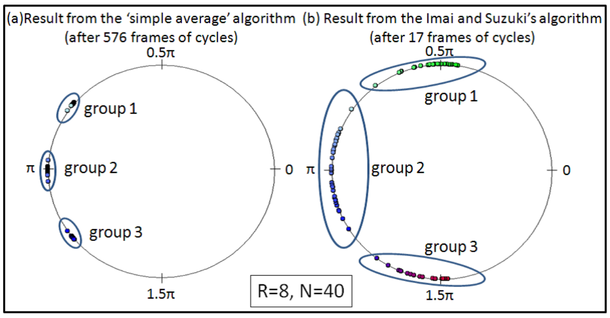


Figure 5: Timing distribution right after the synchronization is realized for three groups within LOS

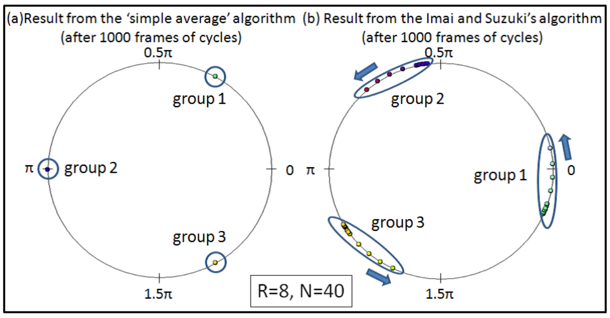


Figure 6: Timing distribution after 1000 frames of cycles, for three groups at the intersection

3.3. Elimination of incomplete synchronization

In the case of three groups at the intersection, above mentioned undesired synchronous patterns; so called mode-lock states are caused by terminal vehicles. The Imai and Suzuki's algorithm avoid this mode-lock state with a kind of the randomly perturbing effect to certain vehicles (; terminal vehicles in this case), but this does not guarantee the remove of mode-lock states as pointed out in [1]. In contrast, it is expected that the Shinohara and Tanaka's algorithm[6] destabilizes such mode-lock states. The idea of their algorithm is certain nodes (; terminal vehicles) temporally quit sending its erroneous timing messages and only receives timing messages from other vehicles. Then, we propose a refined synchronization algorithm to destabilize the mode-lock state by the another simple idea. That idea is each terminal vehicle attain the synchronization with other terminal vehicle which has the nearest timing by priority. It is expected that the synchronization of all groups is totally realized by repeating synchronization for two groups.

3.4. Synchronization ability for these algorithms

Then, to compare different synchronization algorithms, we introduce the degree of synchronization for vehicles. Using phase of timing which is mapped on the unit circle, degree of synchronization for group 1 and 2 σ can be clearly defined as $\sigma(t) = \frac{1}{N_{1,2}} \left| \sum_{i=1}^{N_{1,2}} e^{j\phi_i(t)} \right|$,where $\phi_i(t)$ represents the phase of the i-th vehicle at time t and $N_{1,2}$ is the number of vehicles in group 1 and 2. By this definition, σ becomes 1 when vehicles in group 1 and 2 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 state remains σ always become less than 1.

Below, systematic simulations are carried out to compare the performance for above mentioned algorithms, by using the degree of synchronization σ. The simulation setup follows the setup for the case of three groups in Fig. 1(c). Figure 7 shows σ against time for these algorithms. As shown in Fig. 7, the proposed algorithm realized the perfect synchronization in shorter frames of cycles than the Shinohara and Tanaka's algorithm. Then, we analyze whether proposed algorithm can destabilize mode-lock states or not in more general case. Figure 8 shows the results of 1,000 trials with random initial timings of groups at every trial. Each mark in Fig. 8 represents the result of each trial at the its initial timing difference. As shown in Fig. 8, mode-lock states are eliminated by applying the proposed algorithm.

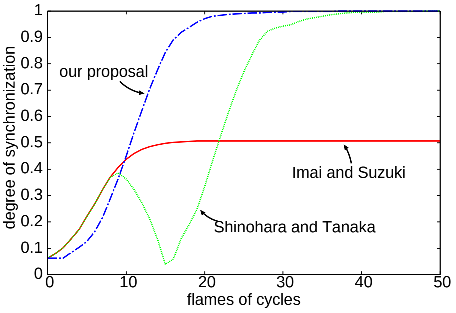
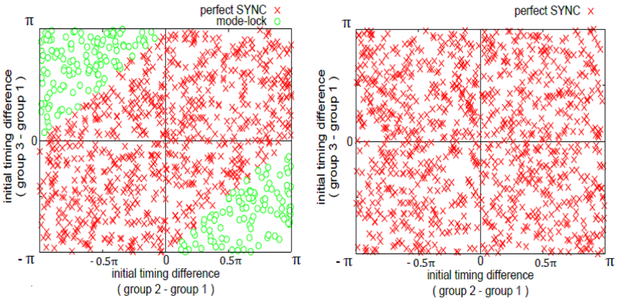


Figure 7: Synchronization ability for these algorithms



(a) Result from the Imai and Suzuki's algorithm

(b) Result from the proposed algorithm

Figure 8: Synchronous pattern distribution for both algorithm

4. Stability analysis of synchronous patterns

In this section, we analyze the stability of (proper or undesired) synchronous patterns. The purpose of our analysis is described as follows.

(i)We focus terminal vehicles which obtained synchronous patterns, to consider the essence of this analysis.

(ii)We put the Imai and Suzuki's algorithm into the mathematical expression, and we examined the linear stability analysis by using this expression.

As a result, we show the proper synchronization is stably obtained for the case of two groups of vehicles, but it is not always obtained for three groups of vehicles. Details of our analysis will be presented in our talk at NOLTA.

5. Conclusions and discussions

Through the systematic simulations and analysis, we clarified the following facts. (i) By using the Imai and Suzuki's synchronization algorithm, two groups of vehicles always realize the proper (perfect) synchronization and this synchronization is stable. (ii) On the contrary, for three groups of vehicles, an undesired synchronous pattern exists and this is stabilized, depending on the initial timings of vehicles.

And, as a byproduct, we constructed a refined synchronization algorithm destabilizing to the proper, perfect synchronization.

Further studies are required to (i) clarify the stability of the proper and undesired synchronization patterns, for the case of general N groups, and (ii) construct a better synchronization algorithm which applies to any N groups.

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

[6] Kenta Shinohara and Hisa-Aki Tanaka, "Mode-Lock Eliminating Timing Synchronization Algorithm for Intervehicle Ad-hoc Networks," *Proceedings of 2008 International Symposium on Nonlinear Theory and Its Applications*, Sep.7–10 2008, pp. 720–723.