Synchronization Inspired from Nature for Wireless Meshed Networks

Alexander Tyrrell Gunther Auer
DoCoMo Euro-Labs
80687 Munich, Germany
Email: lastname@docomolab-euro.com

Christian Bettstetter
University of Klagenfurt, Mobile Systems Group
9020 Klagenfurt, Austria
Email: Firstname.Lastname@uni-klu.ac.at

***Abstract*— In this paper a decentralized synchronization algorithm for ad hoc networks is presented. The algorithm is derived from a model describing synchronization phenomena in biological self-organized systems. Constraints in a wireless environment, such as the fact that a node cannot receive and transmit simultaneously (deafness), the duration of a synchronization message and the delay required at the receiver to decode this message, are taken into account. When applied to meshed networks where nodes have few neighbors, deafness becomes problematic. That is, nodes which are transmitting cannot receive messages, an effect which becomes increasingly severe as the number of neighboring nodes decreases. We propose a delay tolerant synchronization scheme which retains high synchronization rates even in difficult network topologies, with an accuracy only limited by propagation delays.**

I. Introduction

Time synchronization is a classical and important building block in networked systems. Unfortunately, the Network Time Protocol (NTP) and other classical decentralized synchronization protocols are designed and optimized for fixed networks and are not well-suited for wireless meshed networks. In fact, most protocols impose prohibitive constraints when applied to a wireless environment, in particular if the network topology changes over time [1–3].

Approaching this problem, we can learn from nature, more specifically from fireflies in South-East Asia [4]. There, thousands of fireflies gather on trees at dawn and synchronize their blinking, making it seem as though a whole tree is flashing in perfect synchrony. This phenomenon has been known for a long time and can be modeled in a mathematical manner using the theory of pulse-coupled oscillators [5]. An oscillator represents a firefly, and an infinitely short pulse represents an emitted light flash. Each oscillator transmits pulses periodically, and upon reception of a pulse from another oscillator adjusts its clock. Over time, synchronization emerges, i.e. pulses of different oscillators are transmitted simultaneously. Synchronization in populations of coupled oscillators lies within the field of discrete nonlinear dynamics. A theoretical framework for the convergence of synchrony in fully-meshed networks was published in [6]. This work was extended in [7], and the synchronization scheme was mathematically proven to work in meshed networks. In this paper, the term meshed network is used as a synonym for not fully-meshed multi-hop network topologies.

The underlying theory of pulse-coupled oscillators assumes that nodes interact using discrete synchronization pulses that are immediately perceived and interpreted by all other nodes (coupling without delay). In a wireless environment, this network model ignores the fact that communication messages are in general of non-zero duration and some time is required for decoding. To correctly extend the original synchronization scheme (which is reviewed in Section II) to wireless systems, we need to account for two fundamental constraints of typical wireless networks: First, instead of infinitely short pulses, we consider *long synchronization messages*, such as a pseudo-noise (PN) sequences. Furthermore, we allow the nodes to have some processing time when receiving a message. The second constraint is that nodes are *not able to transmit and receive simultaneously*. These constraints result in an inherent delay in the coupling between nodes. To regain high accuracy, a Time Advance Synchronization scheme (TAS) was introduced in our previous work [8].

In [8], it is assumed that the network forms a fully-meshed topology. Motivated by [7] we aim to extend in the present paper the TAS scheme to meshed networks. Unfortunately, the synchrony rate of the TAS scheme decreases when applied to meshed networks.

The contribution of this paper is twofold: in Section III we generalize the TAS scheme in order to adjust the transmitting and receiving periods to a predefined MAC frame structure. In Section IV we allow a node to choose between a transmission time-slot, where the synchronization word together with payload data is transmitted, and a listening time-slot, where its time reference is adjusted and data is received. Thus synchronization in the network is done seamlessly by identifying a synchronization word when in a receiving time slot, and transmitting the same whilst in a payload data time slot. With the generalized synchronization scheme, synchonization of the entire network is always reached. Furthermore, some contraints of the original TAS scheme of [8] on the MAC layer can be relaxed.

II. Synchronization of Pulse-Coupled Oscillators

A. Mathematical Model

As a convenient mathematical representation, each pulse-coupled oscillator i ($1 \leq i \leq N$) in a network of N oscillators ($N > 1$) is modeled by its phase function $\phi_i(t)$. This function evolves linearly over time t until it reaches a threshold value ϕ_{th}. When this happens, the oscillator is said to fire, meaning that it will transmit a pulse and reset its phase. If not coupled to any other oscillator, it will naturally oscillate and fire with a period T. Fig. 1(a) plots the evolution of the phase function during one period when the oscillator is isolated.

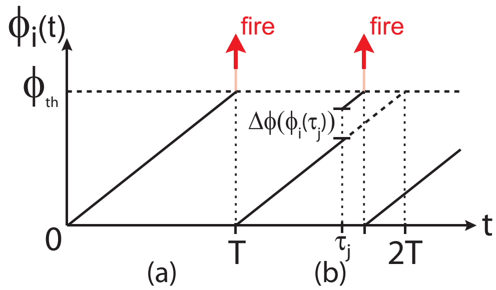

Fig. 1. Time evolution of the phase function

The phase function can be seen as an internal counter that determines when a pulse should be emitted. The goal of the synchronization algorithm is to align all internal counters, so that all nodes agree on a common firing instant. To do so, the phase function needs to be adjusted. In the following, we consider that all nodes have the same dynamics, i.e. clock jitter is considered negligible.

B. Synchronization of Oscillators

When coupled to others, an oscillator is receptive to the pulses of its neighbors. Coupling between nodes is considered instantaneous, and when a node j ($1 \leq j \leq N$) fires at $t = \tau_j$, i.e. $\phi_j(\tau_j) = \phi_{\text{th}}$, all nodes adjust their phase function as follows:

$$\begin{cases} \phi_j(\tau_j) = 0 \\ \phi_i(\tau_j) = \phi_i(\tau_j) + \Delta\phi(\phi_i(\tau_j)) \text{ for } i \neq j \end{cases} \quad (1)$$

Fig. 1(b) plots the time evolution of the phase when receiving a pulse. The received pulse causes the oscillator to fire early. By appropriate selection of $\Delta\phi$, a system of N identical oscillators forming a fully-meshed network is able to synchronize their firing instants within a few periods [6]. The phase increment $\Delta\phi$ is determined by the Phase Response Curve, which was chosen to be linear in [6]:

$$\phi_i(\tau_j) + \Delta\phi(\phi_i(\tau_j)) = \min\left(\alpha \cdot \phi_i(\tau_j) + \beta, 1\right)$$

$$\text{with} \quad \begin{cases} \alpha = \exp(b \cdot \epsilon) \\ \beta = \frac{\exp(b\cdot\epsilon)-1}{\exp(b)-1} \end{cases} \quad (2)$$

where b is the dissipation factor and ϵ is the amplitude increment. Both factors determine the coupling between oscillators, and are identical for all. The threshold ϕ_{th} is normalized to 1.

It was shown in [6] that if the network is fully-meshed, the system always converges, i.e. all oscillators will agree on a common firing instant, for $b > 0$ and $\epsilon > 0$.

This synchronization property is very appealing. Nodes do not need to distinguish between transmitters, and simply need to adjust their internal clock $\phi_i(t)$ by a phase increment when receiving a pulse and transmit a pulse when firing. After some time, synchronization emerges from an initially unsynchronized situation, and pulses are transmitted synchronously.

When delays are introduced in the system, such as propagation delays, a system of pulse-couped oscillators can become unstable, and the system is unable to synchronize [9]. Reducing the coupling helps limiting unstabilities. Another improvement to regain stability is to introduce a refractory period of duration T_{refr} after transmitting. During this period no phase increment is possible [10].

III. Delay Tolerant Firefly Synchronization

The previous section described how time synchronization can be obtained thanks to a distributed algorithm inspired from firefly synchronization. To apply this model to wireless networks, four delays need to be taken into account:

- T_0 Propagation delay: time to propagate from an emitting node to a receiving node. This time is proportional to the distance between two nodes.
- T_{Tx} Transmitting delay: length of the burst. While transmitting, a node is in a `transmit` state and cannot listen to other synchronization messages.
- T_{dec} Decoding delay: time required by the receiver to decode a synchronization message.
- T_{refr} Refractory delay: time necessary after transmitting to maintain stability. A node is in `refr` state during this period.

These delays affect the original scheme in two ways. Firstly, a node follows three states: `transmit`, `refr` and `listen`. The listening state is defined as the period where $\phi_i(t)$ increases from 0 to ϕ_{th} and phase increments by $\Delta\phi$ are possible. Secondly the coupling between nodes is not instantaneous anymore. This condition was necessary for all nodes to be able to fire exactly at the same instant. As the delay in the coupling is now equal to $T_0 + T_{\text{Tx}} + T_{\text{dec}}$, the achievable accuracy is also equal to this duration [8]. Throughout this paper, the propagation delay is considered negligible compared to T_{Tx}, so $T_0 = 0$.

A. Time Advance Synchronization Scheme

In [8], we introduced the TAS scheme to compensate for T_{Tx} and T_{dec}, by introducing a waiting state, denoted by $\texttt{wait}_{\texttt{Tx}}$ before `transmit`, so that $T_{\text{wait,Tx}} = T - (T_{\text{Tx}} + T_{\text{dec}})$. The state machine of the TAS scheme is shown in Fig. 2.

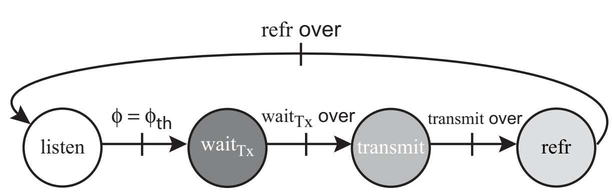

Fig. 2. State machine of the Time Advance Synchronization scheme

Now, when a node j fires at τ_j, i.e. the phase function reaches its threshold, $\phi_j(\tau_j) = \phi_{\text{th}}$, all nodes adjust their phase function exactly T seconds after node j fired:

$$\begin{cases} \phi_j(\tau_j) = 0 \\ \phi_i(\tau_j + T) = \phi_i(\tau_j + T) + \Delta\phi\left(\phi_i(\tau_j + T)\right) \\ \qquad\qquad \text{for } i \neq j \ \& \ i \in \texttt{listen} \end{cases} \quad (3)$$

Nodes i and j are coupled only if a synchronization word is fully received by node i. This effectively means that i needs to be in `listen` state a time $T_{\text{Tx}} + T_{\text{dec}}$ before and after firing. The TAS strategy has two consequences: first, coupling is delayed by one period T, and the intrinsic period of a node equals now $2 \cdot T$. This forms two groups of nodes dynamically, each group firing T seconds after the other [8]. Second, not all nodes are mutually coupled, if $i \notin$ `listen`, no coupling occurs, referred to as *deafness* between nodes. In a fully-meshed topology, mutual deafness between all nodes is prevented by the fact that it is unlikely for a node to be uncoupled to all other nodes. However, when nodes have only few neighbors, mutual deafness becomes more likely, and the network will not be able to synchronize.

B. *Generalized Time Advance Synchronization Scheme*

In the aforementioned TAS scheme, $\texttt{wait}_{\text{Tx}}$ is done exclusively at the transmitter. In order to enhance the flexibility, a waiting state at the receiver $\texttt{wait}_{\text{Rx}}$ of duration $T_{\text{wait,Rx}}$ may be added, in the way that $\texttt{wait}_{\text{Tx}}$ and $\texttt{wait}_{\text{Rx}}$ are done before sending and after receiving the synchronization word, respectively. Then (3) still applies if the following relation is satisfied:

$$T_{\text{wait,Tx}} + T_{\text{Tx}} + T_{\text{wait,Rx}} = T \ , \quad T_{\text{wait,Rx}} \geq T_{\text{dec}} \qquad (4)$$

The constraint $T_{\text{wait,Rx}} \geq T_{\text{dec}}$ is necessary to allow the receiver to decode its synchronization message.

IV. Implications on the MAC Layer

The objective of the synchronization protocol is to get the network to agree on a common time scale. However, the MAC layer imposes additional constraints on the synchronization protocol:

- Given a time-slotted frame structure, slots are decomposed into transmitting and receiving slots of length T.
- A transmit slot typically consists of reference symbols, e.g. a preamble, which may serve as the synchronization word, while the remaining resources are reserved to transmit data.
- The time-slot allocation is dictated by the MAC layer; whether a time-slot is a transmit or receive slot entirely depends on the chosen MAC protocol.

The generalized TAS scheme can be cast into such a time-slotted frame structure by defining a Transmitting Period and a Receiving Period, both of duration T. This decomposition is shown on Fig. 3.

A transmitting time slot is composed of three states. A transmitter stays in $\texttt{wait}_{\text{Tx}}$ during $T_{\text{wait,Tx}}$ before transmitting the synchronization word. Then during `transmit`, the synchronization word of duration T_{Tx} is transmitted. Finally a transmitter goes into $\texttt{wait}_{\text{Rx}}$ during $T_{\text{wait,Rx}}$. Payload data can be transmitted during $\texttt{wait}_{\text{Rx}}$ and $\texttt{wait}_{\text{Tx}}$. The state `transmit` now corresponds to the transmission of a synchronization word. $T_{\text{wait,Tx}}$ can be set to zero if the synchronization word is the preamble. If the synchronization sequence is in the middle of a burst, such as in the GSM standard, $T_{\text{wait,Tx}} = T_{\text{wait,Rx}}$.

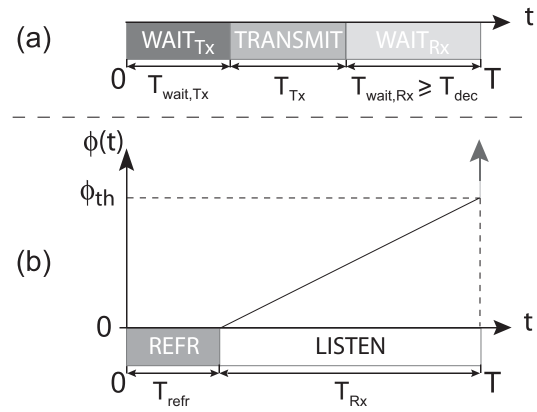

Fig. 3. Separation of the states composing the Time Advance scheme into (a) a Transmitting Time Slot and (b) a Receiving Time Slot

The receiving period is composed of two states: `refr` and `listen`. To match the receiving period to the slot duration we get $T = T_{\text{refr}} + T_{\text{Rx}}$. At the beginning of a Receiving Period, a node will stay in the refractory state during a time equal to T_{refr}. While in refractory state, a node switches on its receiver, however, no phase increment is possible. During `listen`, for T_{Rx} seconds the phase function is adjusted when a synchronization message is received. The data conveyed in the remaining part of the slot may be processed in parallel.

Two allocation schemes are considered, to decide whether the next slot is a transmit or receive slot:

- Alternating receive and transmit time-slots
- Random slot allocation

The alternating allocation scheme is equivalent to the original TAS scheme described in Section III-A. This corresponds to an allocation where a transmit slot is always followed by a receive slot. However, the decision whether to receive or transmit data should be left to the scheduling policy of the MAC protocol, e.g. dependent on whether there is data to transmit in the buffer, or if other nodes request to transmit data. Since the scheduling policy may appear random to the synchronization unit, random slot allocation may model the scheduling policy of the MAC protocol more realistically.

V. Application to Meshed Networks

The performance of the TAS scheme is evaluated through computer simulations. To verify the aforementionned deafness effect, it is instructive to study the TAS strategy with a worst case topology. As deafness depends heavily on the number of neighbors, it will be most present when a node has only one or two neighbors. Therefore a line topology of eight nodes, shown on Fig. 4, is selected. The TAS scheme is applied as T_{Tx} varies

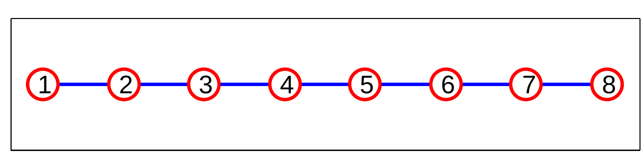

Fig. 4. Considered line topology of 8 nodes

from $0.1 \cdot T$ to $0.5 \cdot T$. Sets of 1000 initial conditions were generated for each value of T_{Tx} using a program written in Matlab. The initial conditions correspond to the case where all nodes have randomly distributed state variable, i.e. each node is assumed to be active starting with a random phase $\phi_i(0)$. Each simulation runs for $80 \cdot T$ and if synchrony is not reached at the end of this period, synchrony is declared unsuccessful. Each period T is decomposed into 1500 steps, and at each step, state and interactions of each node are evaluated. Delays are fixed to $T_{\text{dec}} = 0.1 \cdot T$ and $T_{\text{refr}} = 0.3 \cdot T$, and the coupling parameters from (2) are set to $\alpha = 1.3$ and $\beta = 0.02$, which satisfy the conditions of [6].

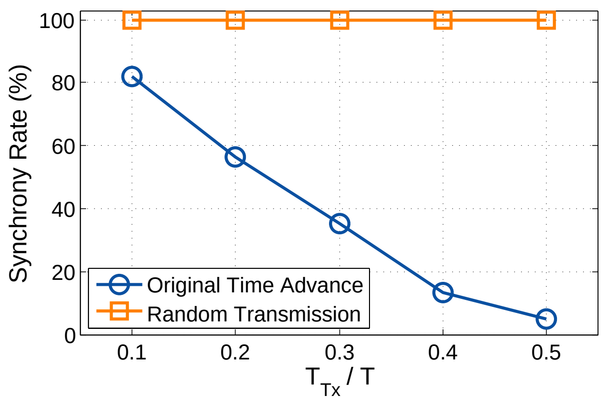

Fig. 5. Synchronization results when applying the Generalized Time Advance Synchronization scheme

Results for the synchronization rate are shown on Fig. 5. For alternating slot allocation, the degradation of the synchrony rate as the transmission time T_{Tx} increases confirms that deafness during transmission is responsible. A synchrony rate lower than 15% when the transmission duration is higher than $0.4 \cdot T$ is not acceptable. Random slot allocation, on the other hand, is able to achieve 100% synchrony

This difference is inherently linked to the deafness between nodes. The formation of at least two groups of nodes is a fundamental requirement for synchronization, and this comes from the fact that a node cannot listen while transmitting. Hence it cannot hear nodes that are in its group, and needs to rely on others to adjust its time reference. With the alternating slot allocation scheme, nodes alternatively transmit and receive, and never change group. This becomes problematic in meshed network when a node has few neighbors as, depending on initial conditions, two neighbors can transmit such that their transmit slots overlap. Hence they cannot synchronize. Through the random switching, a node will dynamically change group, and over time it becomes more and more unlikely that transmit slots of neighboring nodes always overlap. While the probability for deafness at a particular period T is as high as for alternating slot allocation, the random switching of transmitting and receiving ensures that deafness is diminishing over time. This shows that synchrony is regained seamlessly, and makes the application of the new scheme very appealing for ad hoc networks.

Fig. 6 plots the mean time to synchrony resulting from the

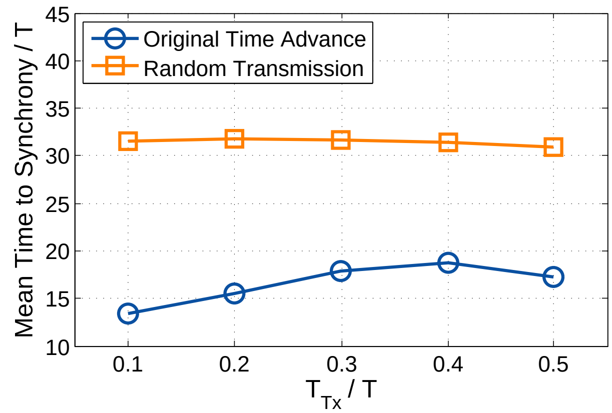

Fig. 6. Mean time to synchrony when applying the Modified Time Advance Synchronization scheme

same simulations as in Fig. 5. Although the network of Fig. 4 has a maximum hop count of 7, the time to synchrony with the random allocating scheme achieves synchrony with an average of $32 \cdot T$, which is about twice as much as for the other scheme.

VI. Conclusion

Thanks to the flexibility introduced by the Generalized Time Advance scheme, time synchronization using long bursts based on the theory of pulse-coupled oscillators is now done seamlessly, and yields encouraging results in meshed topologies. Synchronization in meshed networks is regained by introducing randomness into the original synchronization scheme. In this way, we are helped by the MAC layer, which often dictates random transmissions of packets in decentralized networks. Combining this property with the simple rules of firefly synchronization result in a powerful synchronization scheme.

References

[1] D. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on Communications*, vol. 39, pp. 1482–1493, Oct. 1991.

[2] K. Römer, "Time and location in sensor networks," in *Proc. GI/ITG Workshop on Sensor Networks, Berlin, Germany*, pp. 57–60, July 2003.

[3] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proc. Operating Systems Design and Implementation 2002, Boston, USA*, Dec. 2002.

[4] S. Strogatz and I. Stewart, "Coupled oscillators and biological synchronization," *Scientific American*, vol. 269, pp. 68–74, Dec. 1993.

[5] A. Winfree, "Biological rhythms and the behavior of populations of coupled oscillators," *J. Theor. Biol.*, vol. 16, pp. 15 – 42, Jul. 1967.

[6] R. Mirollo and S. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM J. Appl. Math*, vol. 50, pp. 1645–1662, Dec. 1990.

[7] D. Lucarelli and I. Wang, "Decentralized synchronization protocols with nearest neighbor communication," in *Proc. SenSys'04, Baltimore, USA*, pp. 62 – 68, Nov. 2004.

[8] A. Tyrrell, G. Auer, and C. Bettstetter, "Firefly synchronization in ad hoc networks," in *Proc. MiNEMA Workshop 2006*, Feb. 2006.

[9] U. Ernst, K. Pawelzik, , and T. Geisel, "Synchronization induced by temporal delays in pulse-coupled oscillators," *Physical Review Letters*, vol. 74, pp. 1570 – 1573, Feb. 1995.

[10] Y.-W. Hong and A. Scaglione, "Time synchronization and reach-back communications with pulse-coupled oscillators for UWB wireless ad hoc networks," in *Proc. IEEE Conference on Ultra Wideband Systems and Technologies 2003*, pp. 190–194, Nov. 2003.