Alexander Tyrrell, Diplôme d'Ingénieur

FIREFLY SYNCHRONIZATION IN WIRELESS NETWORKS

DISSERTATION

zur Erlangung des akademischen Grades
Doktor der technischen Wissenschaften

Universität Klagenfurt
Fakultät für Technische Wissenschaften

1. Begutachter:	Univ.-Prof. Dr.-Ing. Christian Bettstetter
Institut:	Institut für Vernetzte und Eingebettete Systeme
2. Begutachter:	Univ.-Prof. Dr. rer. nat. Rudolf Mathar
Institut:	Lehrstuhl für Theoretische Informationstechnik
	Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen Deutschland

Juli 2009

Ehrenwörtliche Erklärung

Ich erkläre ehrenwörtlich, dass ich die vorliegende Schrift verfasst und die mit ihr unmittelbar verbundenen Arbeiten selbst durchgeführt habe. Die in der Schrift verwendete Literatur sowie das Ausmaß der mir im gesamten Arbeitsvorgang gewährten Unterstützung sind ausnahmslos angegeben. Die Schrift ist noch keiner anderen Prüfungsbehörde vorgelegt worden.

Word of Honour

I honestly declare that the thesis at hand and all its directly accompanying work have been done by myself. Permission has been obtained for the use of any copyrighted material appearing in this thesis and all such use is clearly acknowledged. The thesis has not been presented to any other examination board.

Unterschrift: _____

Munich, Germany, 9. Juli 2009

Contents

List of Tables

List of Figures

Acknowledgments

Dear reader,

Thank you for taking the time to pick up this thesis and turning the first few pages. I sincerely hope that you will continue reading this work and that you will find it interesting.

Before diving into the details of firefly synchronization, I would like to express my profound gratitude to my two supervisors, Dr. Gunther Auer and Prof. Christian Bettstetter, the two minds behind the inspiration of applying firefly synchronization to wireless networks. Thank you for steering my work during the last three years. Things would have not gone as well as they did without your supervision and guidance.

I highly appreciated working within a team of experts in wireless systems. I would like to thank my officemate Dr. Tetsushi Abe and the rest of my colleagues from the Wireless Technologies Research group, Dr. Gerhard Bauch, Dr. Guido Dietl, (soon-to-be Dr.) Katsutoshi Kusume, and Andreas Saul, and our head of lab Dr. Stefan Kaiser for their continuous encouragements and support. Thank you for broadening my horizons and knowledge through discussions, project meetings, and colloquiums. I would also like to thank Denis Maligin for the excellent IT support. Running simulations would have been difficult without your help.

I am very grateful to have performed this project within DOCOMO Euro-Labs. I would like to thank Dr. Toru Otsu and Dr. Hendrik Berndt for sustaining such a creative environment. DoCoMo Euro-Labs is certainly at the edge of research in telecommunications, and it has been an immense honor to have been integrated into your company.

I would like to thank my fellow PhD candidates from the Institute of Networked and Embedded Systems, and in particular from the Mobile Systems Group, Helmut Adam, Günther Brandner, Sérgio Crisóstomo, Dr. Wilfried Elmenreich, Michael Gyarmati, Nikolaj Marchenko, and Udo Schilcher. Although I did not spend a long time at the university, you always made me feel welcome and part of the team. Your research work helped me gain perspective on mine.

During the last three years, I have been given the chance to collaborate with some excellent researchers. I would like to thank Dr. Luca Sanguinetti for our work in the multicarrier synchronization project, Prof. Harald Haas for our collaboration on the busy-burst project, and the people from the WINNER II project, in particular the people from the task 3 and WP2. I learned a lot from you all.

Finally, last but certainly not least, I would like to thank very warmly Sophie and my two

daughters Maya and Inès for keeping my feet on the ground throughout the last years. Life is not just about firefly synchronization, and I am happy to have been able to balance life at work and life at home the way I did thanks to you.

Abstract

A striking example of self-organization in nature occurs every evening in some parts of South-East Asia: thousands of fireflies gather on trees at dawn and start emitting flashes regularly; over time, synchronization emerges from a seemingly chaotic situation, which makes it seem as though the whole tree is flashing in perfect synchrony. This fascinating phenomenon is the inspiration for the topic treated in this thesis, which is concerned with slot synchronization in wireless networks.

Synchronization phenomena in nature are mathematically described by the theory of pulse-coupled oscillators (PCOs); each entity naturally oscillates and blinks periodically, and coupling is performed through the discrete emissions of light. Each node adjusts its internal reference when perceiving blinks from its neighbors, and following simple rules, synchronization always emerges after some time. Conditions for convergence under ideal assumptions were derived by Mirollo and Strogatz in their seminal work published in 1990, and provide a framework for the following slot synchronization studies.

The PCO synchronization rules are remarkably simple and robust, which makes their application to wireless networks very appealing. In particular, slot synchronization requires nodes in the network to agree on a common time reference for the start of a slot, in a similar way to fireflies that agree on a common blinking instant. Direct application of the PCO rules is not feasible, and an adaptation, termed Mobile Emergent Firefly Synchronization (MEMFIS), is proposed so that constraints of wireless networks are integrated with the PCO rules. With this modification, the simplicity and robustness of the PCO scheme is retained; nodes are able to synchronize starting from any random misalignment, and achieve an accuracy equal or lower to the direct propagation delay.

Application of the PCO model to cellular systems is investigated. The goal is to maintain base stations synchronized, even when there is no direct communication between them. Synchronization in Cellular Firefly Synchronization (CelFSync) is performed by letting some selected user terminals participate in the network synchronization process, and achieving an out-of-phase synchronization regime. Furthermore propagation delays, which are problematic in large-scale networks, are mitigated by combining the proposed adaptation with the timing advance procedure, so that an acceptable inter-base station accuracy is achieved.

Zusammenfassung

Ein beeindruckendes Beispiel für Selbstorganisation in der Natur kann abends in Teilen Südostasiens beobachtet werden: tausende auf Bäumen sitzender Glühwürmchen (engl. *Fireflies*) emittieren in regelmäßigen Abständen Lichtimpulse. Aus dem anfänglich chaotischen Blinken einzelner Glühwürmchen formiert sich im Laufe der Zeit ein im perfekten Gleichtakt blinkender Baum. Die Methodik dieses faszinierenden Naturschauspiels bildet die Grundlage für das in dieser Dissertation behandelte Thema der Slot-Synchronisation in drahtlosen Netzwerken.

Phänomene natürlicher Synchronisation können durch die Theorie der impulsgekoppelten Oszillatoren (PCOs) mathematisch beschrieben werden. Jeder Knoten eines Netzwerks blinkt in regelmäßigen Abständen, und benachbarte Knoten sind durch die diskrete Emission von Pulsen (z.B. von Licht) miteinander gekoppelt. Einfache Regeln bestimmen, wie die interne Zeitreferenz eines jeden Knotens durch die Detektion von Pulsen beeinflusst wird. Mirollo und Strogatz haben in ihrer grundlegenden, im Jahr 1990 veröffentlichten Arbeit gezeigt, dass ein Netzwerk von PCOs unter bestimmten idealisierten Voraussetzungen immer zu einem Zustand konvergiert, bei dem alle Knoten im Gleichtakt blinken. Die PCO Synchronisation nach Mirollo und Strogatz ist bemerkenswert einfach und robust, was insbesondere für Anwendungen bezüglich der Slot-Synchronisation von dezentralen *ad hoc* Netzen attraktiv ist.

In ähnlicher Weise wie bei im Gleichtakt blinkender Glühwürmchen, ist das Ziel der Slot-Synchronisation, dass alle Knoten eines Netzwerkes sich auf einen gemeinsamen Zeitpunkt einigen, welcher den Beginn eines Zeitschlitzes (engl. *time-slot*) zur Übertragung eines Datenpakets bestimmt. Bedingt durch die spezielle Problematik der drahtlosen Datenübertragung, ist eine direkte Anwendung der PCO Synchronisation für drahtlose Netzwerke ohne weiteres nicht möglich. Insbesondere Zeitverzögerungen, die zwischen dem Senden und dem Empfang eines Synchronisationswortes auftreten, können zu einer signifikanten Verminderung der Genauigkeit und im schlimmsten Falle zu Instabilität führen. Verzögerungen werden verursacht durch die Laufzeit elektromagnetischer Wellen, aber auch durch die Einführung eines Synchronisationswortes endlicher Länge anstelle eines Pulses von infinitesimal kurzer Dauer. In der vorliegenden Arbeit werden Wege aufgezeigt, wie die PCO-Synchronisation den speziellen Anforderungen der drahtlosen Datenübertragung angepasst werden kann, ohne deren Einfachheit und Robustheit zu beeinträchtigen. Simulationen zeigen, dass eine zuverlässige Netzwerksynchronisation für zufällige Anfangsbedingungen mit einer Genauigkeit erreicht wird, deren obere Schranke durch die Laufzeit einer elektromagnetischen Welle zwischen zwei direkt benachbarten Knoten gegeben ist.

Desweiteren wird die Anwendung der PCO Synchronisation auf zellulare Systeme mit dem Ziel untersucht, die Rahmenstruktur von Basisstationen zu synchronisieren, selbst wenn benachbarte Basisstationen keine direkte Verbindung haben. Die Synchronisation erfolgt durch Einbeziehung einiger ausgewählter Nutzer-Terminals. Außerdem wird ein Verfahren vorgestellt, welches die Ungenauigkeiten bezüglich Laufzeitverzögerungen deutlich reduziert, was insbesondere für große Zellen von Bedeutung ist.

CHAPTER

1 Introduction

The word ‘synchronization’ refers to the act of making synchronous different events so that they occur at the same time. Etymologically it is composed of the Greek words ‘syn’ meaning together and ‘khronos’ meaning time. Synchronization in daily life arises in a variety of situations, from people arriving simultaneously at a meeting to dancers or swimmers following the exact same sequence of movements in perfect unison. These displays of perfectly coordinated movements are often fascinating and seem to resonate within ourselves, striking a profound chord.

In telecommunication networks, synchronization is required at many levels and its definition depends heavily on the level of abstraction; it ranges from point-to-point synchronization, which is needed at the receiver to recover the timing of the transmitter, to data synchronization, which enables a user to synchronize its data from different points across the network. The separation into levels of abstractions is needed to simplify the synchronization problem, focusing on key issues and ignoring unnecessary details. This requires to clearly define each synchronization form that is addressed and the problems specific to each.

The problem addressed in this thesis is *slot synchronization*, which is a form of network synchronization. Network synchronization in communications deals with the alignment in time and frequency of clocks placed at different locations. In slot synchronization, time is divided into slots, i.e. intervals of fixed duration, and nodes agree on a common slot start. This is an essential building block in wireless communications and networking. It enables time coordination between nodes, which is necessary in various layer functions such as slotted Medium Access Control (MAC), distributed sensing, scheduling of sleep phases, and cooperative diversity. Furthermore slot synchronization is typically a benefit when it is not a requirement. For instance, slotted MAC protocols offer a higher throughput and enhanced interference protection over unslotted protocols.

Network synchronization is needed because clocks are inherently imperfect. It is thus not sufficient to synchronize them once and some synchronization mechanism needs to be put in place to maintain them aligned in time and frequency. A straightforward approach for a clock to synchronize is to periodically copy the time from a reference clock. Common issues in this case are how to access and transfer timing information from the master clock to the slave clock, or how to estimate and compensate for the delay between the two distant nodes.

An alternative approach to the network synchronization problem is to let nodes self-organize and *agree* on a common timing. In communication networks, self-organization is present at many levels [PB05]; it has been successfully applied to minimize the need for configuration of network addresses, it facilitates some network operations such as routing and addressing; and furthermore it enables new types of communications networks, such as completely decentralized ad hoc and sensor networks. Recently the research interest in ad hoc and wireless sensor networks saw an increasing trend in the development of self-organized algorithms as they provide a number of advantages over centralized approaches including an inherent support for adaptive operations, robustness against changing environmental conditions, and an optimized scalability thanks to simple local rules.

Self-organized systems are characterized by an ensemble of individual units that interact locally with neighboring units to produce a higher-level pattern, e.g. synchronization. Units follow a set of simple rules, and influence directly only their neighbors. No leader, hierarchy nor pacemaker units are present, and the emergent behavior is produced only through these local interactions. Designing a self-organized algorithm therefore necessitates to give up centralized control and to take a bottom-up approach in order to obtain the desired emergent higher-level pattern. This form of design is sometimes difficult to grasp, because the connection between local and global behavior may differ. Furthermore it is difficult to think in terms of decentralized ensembles, given the multiplicity of links and interactions all occurring simultaneously.

Facing this paradigm shift, Nature is a formidable source of inspiration for designing self-organized algorithms. Many biological systems are governed by a set of rules that transcend the simplicity of local units. A clear example of this is the formation of organs in the body. Each organ is composed of a large number of cells. Initially a new cell, called a stem cell, does not have any particular function. At the differentiation stage, this stem cell specializes, and its function is decided both by its internal program, e.g. the DNA, and by its direct environment, e.g. neighboring cells. Self-organization principles further explain the formation of industrial neighborhoods along class lines, the building of nests and the foraging of ants, or the pattern recognition and cognition capabilities of the brain [Joh01].

Another fascinating example of self-organization in Nature is the ability of fireflies to synchronize their flashing instant [Buc88, Str03]. For this self-organized synchronization, local rules can be summarized as follows: each firefly maintains an internal timer determining its flashing instant, and updates it based on the flashes from neighboring fireflies. Section 1.1 presents the basic mechanism behind firefly synchronization. These experiments were made on real fireflies, and are helpful to understand the local rules behind self-organized synchronization. Despite the simplicity of the local synchronization dynamics, it took many years to develop a formal proof. In 1990 Mirollo and Strogatz published a seminal work that mathematically proved under which conditions synchronization is always reached in a self-organized manner [MS90]. Since this proof, the idea of applying the firefly synchronization rules has attracted the attention of researchers in various fields such as neural science, imaging, or wireless communications.

Before detailing network synchronization, Section 1.2 summarizes some discussions among

philosophers and mathematicians on the notion of time. Given the etymology of the word synchronization, time is a fundamental principle underlying this thesis, but no clear agreement on its definition is available. This chapter further identifies the goals and advantages of network slot synchronization. A detailed outline and summary of principal contributions of this thesis are presented in Section 1.3. The contributions are spread over four main chapters, and concern the definition of network slot synchronization (Chapter 2), the original firefly synchronization model (Chapter 3), an adaptation to wireless decentralized networks (Chapter 4), and an application to perform self-organized synchronization in cellular networks (Chapter 5).

1.1 Firefly Synchronization

On riverbanks in South-East Asia male fireflies gather on trees at dawn, and emit flashes regularly. Over time, synchronization emerges from a seemingly chaotic situation, which makes it seem as though the whole tree is flashing in perfect synchrony. This phenomenon forms an amazing spectacle, and has intrigued scientists for several hundred years [Buc88]. Over the years, two fundamental questions have been studied. Why do fireflies synchronize? And how do they synchronize?

The first question led to many discussions among biologists. In all species of fireflies, emissions of light serves as a means of communication that helps female fireflies distinguish males of its own species. While male as well as female fireflies emit light, the response of male fireflies to emissions from females is different in each species. For some species such as *Pteroptyx cribellata* and *Pteroptyx malaccae*, males synchronize their flashing. The reason behind this spontaneous synchronization remains subject of controversial discussions. Several hypothesis exist: either it could accentuate the males' rhythm or serve as a noise-reduction mechanism that helps them identify females [OS77]. This phenomenon could also enable small groups of males to attract more females, and act as a cooperative scheme.

Early hypotheses had difficulties explaining the firefly synchronization phenomenon. For example, Laurent in 1917 dismissed what he saw and attributed the phenomenon to the blinking of his eyelids [Lau17]. Others argued that synchrony was provoked by a single stimulus received by all fireflies on the tree [Bla15]. However the presence of a leading firefly or a single external factor is easily dismissed by the fact that not all fireflies can see each other and fireflies gather on trees and progressively synchronize. The lack of a proper explanation lasted until the 1960s and is mostly due to a lack of experimental data.

Among early hypotheses, Richmond [Ric30] stated in 1930 what came very close to the actual process:

> Suppose that in each insect there is an equipment that functions thus: when the normal time to flash is nearly attained, incident light on the insect hastens the occurrence of the event. In other words, if one of the insects is almost ready to flash and sees other insects flash, then it flashes sooner than otherwise. On the foregoing hypothesis, it follows that there may be a tendency for the insects to fall in step and flash synchronously.

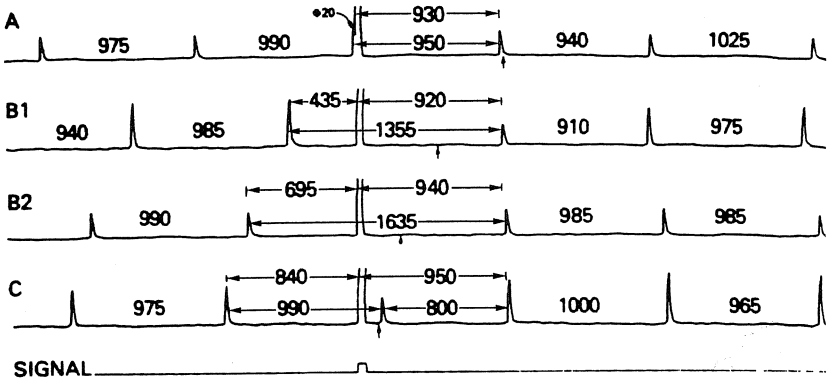


Figure 1.1: Experiments by Buck *et al.* (from [BBCH81] with kind permission of ©Springer Science and Business Media). Delays are expressed in ms.

This statement identifies that synchronization among fireflies is a self-organized process, and fireflies influence each other: they emit flashes periodically, and in return are receptive to the flashes of other fireflies.

To understand the adaptation of a firefly's rhythm to external pulses, Buck *et al.* [BBCH81] identified the reaction of fireflies to an external signal. These experiments concentrated on the reaction of a firefly to an external signal depending on *when* this light is received. Naturally a firefly emits light periodically every 965 ± 90 ms [BBCH81], and the external signal changes this natural period. For the experiments, the firefly was put in a dark room and was restrained from seeing its own flashes. Stimuli were made by guiding 40 ms signals of light from a glow modulator lamp into the firefly's eye via a fiber optics. Responses were recorded and are shown in Figure 1.1.

From Figure 1.1, when an external signal is emitted (bottom line in the figure), three different responses are identified:

- In response A, the artificial signal occurs only 20 ms after the firefly's spontaneous flash. As the following response from the firefly occurs at a normal time of 950 ms, the signal has not modified the natural response. This behavior corresponds to a refractory period: during this time, the potential of the flash regains the "resting" position and no modification of the internal clock is possible.

- In responses B1 and B2, the signal inhibits the response of the firefly: instead of emitting light after about 960 ms, it delays its response until 920 ms and 940 ms after receiving the signal. Thus successive flashes occur 1355 ms and 1635 ms apart, which is far more than the natural period.

- In response C, the artificial signal occurs 150 ms before the natural flashing, and does not have any incidence on this flash. This is due to a processing delay in the central nervous system of a firefly, which is equal to about 800 ms [BBCH81]. Therefore the external signal influences the following flash, which is advanced by 150 ms. The external signal thus exhibits an excitatory effect on the response and brings the firefly to flash earlier.

In all responses the firefly synchronizes with the external signal. The modified behavior of the firefly depends only on the instant of arrival of the external signal, and this flash altered only the emission of one following flash, i.e. in the following period nodes regained their natural period of about one second. Variating the amplitude of the input signal yielded similar results. The responses display excitatory coupling (responses B1, B2, and C), and a refractory period placed after emission is present (response A). Interestingly in responses B1, B2, and C, the firefly delayed its adaptation by one period to allow for a processing delay, and synchronization is achieved at the following period.

For more insights into the phenomenon of firefly synchronization, Chapter 10 in [CDF+01] provides a history of studies on fireflies, including early interpretations, and analyzes different experiments including the one presented in this section.

These experiments have helped mathematicians modeling fireflies. In 1990 a thorough mathematical model deriving synchronization among an arbitrarily large network of coupled oscillators was derived by Mirollo and Strogatz [MS90]. The rules governing this model are intriguingly simple:

- each node maintains a periodic function that acts as an internal reference defining flashing instants;
- when perceiving an external flash, the function is adjusted based on coupling parameters and the current value of the internal function.

Provided that certain constraints on the coupling between oscillators are met, Mirollo and Strogatz proved that, for an arbitrary number of nodes and independent of the initial conditions, the network always synchronizes [MS90]. For simplicity this model considers instantaneous excitatory coupling and no refractory period. This model forms the basis for the adaptations presented in this thesis, but the proposed adaptations further integrates a processing delay into the local rules, so that couplings are performed at the next period.

1.2 Synchronization in Communication Networks

1.2.1 Time

Underlying the studies on synchronization lies the notion of time. Time plays a central role in our lives, and this has been so since the beginning of civilization, e.g. to plan crop harvest. The debate around the notion of time is almost as old, and spawned many discussions among mathematicians and philosophers. Among the latter, at the end of the 4th century St. Augustine is famous for his quote on the difficulty of defining time [Aug38]:

> What then is time? If no one asks of me, I know; if I wish to explain to him who asks, I know not.

Defining and characterizing time has been a topic of many discussions among philosophers since Ancient Times. Among the first to debate on the notion of time, Zeno of Elea, around 450 BC, presented a series of paradoxes to show that motion is an illusion. His third paradox, known as the arrow paradox, states, in Aristotle's words [Ari99]:

> The third is (...) the flying arrow is at rest, which result follows from the assumption that time is composed of moments (...). [Zeno] claims that if it is always true that a thing is at rest when it is opposite to something equal to itself, and if a moving object is always in the now, then a moving arrow is motionless.

Thus, for Zeno, the arrow cannot move because it does not move in the considered instant. Later, around 350 BC, Aristotle argued that if time is not composed solely of nows, i.e. of discrete instants as opposed to segments, then the argument of Zeno is not valid. For Aristotle, time itself is motion: the flow of time is evident in the motion of the arrow. Part of this disagreement and the difficulty in agreeing on what time is lies in the use of the single word 'time' to denote two distinct concepts. The first is the indication of an event, a single point in time. The second is an interval or a duration between two events [JFR99].

Other debates on the notion of time are concerned with the origin of time (Plato stated that time began with the universe), its structure, and its existence. Regarding its structure, time can be represented as a straight arrow composed of two large blocks, the past and the future, that are separated by a thin line that is the present. In contrast, time can be seen as cyclic; years, months, days, hours repeat themselves in a periodic fashion. A fascinating example of this is visible in the cyclic calendars built by the ancient Mayan civilization.

Despite the difficulty of defining and agreeing on the notion of time, in 1687 Newton was the first to give a *mathematical* definition of time [New48]:

> Absolute, true, mathematical time, of itself, and from its own nature flows equably without regard to anything external, and by another name is called duration: relative, apparent, and common time, is some sensible and external (whether accurate or unequal) measure of duration by the means of motion, which is commonly used instead of true time; such as an hour, a day, a month, a year.

This was a revolution in the way mathematicians thought about time: time is no longer determined by the universe but it is given by an external and absolute clock. This absolute time defined by Newton is usually denoted by t and is key in the laws of motion and in the studies of dynamical systems, among others. This 'classical' definition of time was later opposed to the 'relativistic' definition given by Einstein who related time with space in the special relativity and general relativity theories, stating that time depends on the spatial reference of the observers. For the remainder of this thesis, the classical definition of time is assumed.

Along this mathematical definition and the debate on the notion of time, techniques were developed for keeping time. Sundials and water clocks were almost the only available devices used until the 14th century, until the first mechanical clocks were built. These were very large (over ten meters

high) and constituted real works of art, both in terms of their appearance but also because of their clockwork and of the difficulty of making them work. In the 17th century, Galileo discovered that pendulum could be used for accurate time keeping, but it was Huygens who built the first pendulum clock in 1656.

The problem of synchronizing clocks located at different locations did not become important before the 18th century and was triggered by ship navigation. Before this, only latitude could be determined thanks to a sextant, and no solution existed for determining longitude from a ship. Time and longitude are directly related by the rotation of the Earth, i.e. 360° per day, and longitude calculation is computed by comparing the apparent local time and the time kept on the clock synchronized with the known location. In 1759 John Harrison's H-4 clock won the "longitude prize" by erring by only 5 seconds over an 81 day journey, corresponding to 1.5 miles in equatorial latitude [Win00]. Later, with the advent of the train in the 19th century, the definition of a global international time reference became necessary so that railway stations would agree on departure and arrival times, and people would not miss their train. After the Second World War, with the development of intercontinental communications and advances in technologies of transportation and navigation, timekeeping, and space tracking, the requirements on time and frequency information became more stringent [Bla73]. This required efforts in researching and developing network synchronization techniques.

1.2.2 Network Synchronization

Network synchronization deals with the agreement of clocks in time and frequency placed at different locations. Clocks are unavoidably imperfect, implying that even if all agree on the same time and frequency at a given time, they lose track of time over time and therefore need to synchronize periodically.

Clocks are subject to several imperfections. They are composed of a counter, which maintains the value of time, and an oscillator, which updates the counter at each tick. The time is read by referring to the counter value, and therefore time synchronization is the agreement on a common clock value among the network participants. Frequency is measured by the period of the oscillator, i.e. the interval of time between two consecutive ticks, and frequency synchronization is defined as the agreement on a common interval duration.

Turning to wireless networks, classical means of providing a common time and frequency involves centralized control, where all nodes in the network follow the time reference given by a master node, which broadcasts a reference signal (beacon). In wireless networks, some systems such as Bluetooth [IEE03] and Zigbee [IEE05] rely on time synchronized medium access only when the beacon can be heard within the entire network; otherwise the system resorts to an asynchronous fallback mode. Clearly, in ubiquitous computing systems, where wireless technologies interconnect a large number of heterogeneous devices to multi-hop networks, centralized approaches are infeasible. This thesis investigates a decentralized solution to the network slot synchronization problem.

1.2.3 Slot Synchronization

Slot synchronization is a special form of network synchronization. It requires nodes to agree on a common time slot structure: time is divided into fixed intervals termed slots, and nodes should agree on a common beginning and end of a slot. In contrast to time synchronization, which requires nodes to agree on a common time value, slot synchronization requires nodes to agree on the remaining time until the next slot start. This is a fundamental difference in the approach to network synchronization.

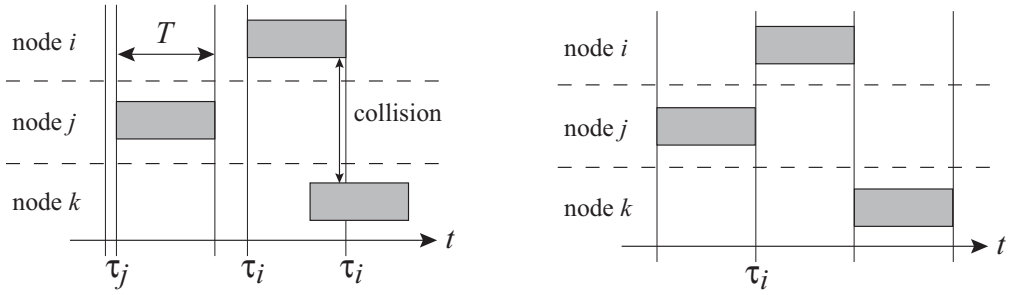
Slot synchronization is a form of coordination between nodes that can be used implicitly, and it is usually assumed that it is available. While asynchronous protocols avoid slot synchronization, slot synchrony typically brings benefits over non-synchronized systems and is often a requirement for applying certain techniques and algorithms on the physical and MAC layers (see for examples [CGL00, SdV04, TG06, CCS06, KVM$^+$07, OHA07]). It enables and facilitates:

- the use of advanced transmission technologies. One example is the use of distributed antenna systems, where nodes transmit the same signal simultaneously, leading to a spatial diversity gain;
- the deployment of slotted MAC protocols. This is exemplified for the ALOHA protocol below;
- a simpler design of frequency hopping schemes and of interference coordination and management techniques as the transmission of a node does not overlap over two slots [KXC$^+$00, WH00];
- in cellular systems, when a mobile user moves from its attached cell to another and requires a handover, slot synchronization enables fast cell search [JSL02].

Two illustrations of the mechanism and benefits of slot synchrony are further detailed below. The first example discusses the ALOHA protocol, which is at the heart of the Internet Protocol (IP); slot synchronization in this case doubles the maximum achievable throughput. The second example discusses the scheduling of sleep periods, which requires synchronization among nodes.

Slotted ALOHA

The ALOHA protocol [Abr77] provides the key functionality of distributed random access in wireless systems. Packets are instantly transmitted whenever they arrive at the MAC layer. If a packet collides at the receiver with a packet transmission from another node, it will be retransmitted after a random backoff period. The synchronized variant of ALOHA is Slotted-ALOHA (S-ALOHA). Here, nodes are only allowed to transmit at the beginning of a slot. For a given traffic load, the channel utilization of S-ALOHA is always higher than that of ALOHA, and the maximum channel utilization is doubled from 18 % to 36 % [Abr77]. The basic functionality of S-ALOHA is an important ingredient in many of today's random access protocols (e.g. in IEEE 802.11). Moreover, slot synchronization is an enabler for reservation-based medium access, such



(a) Asynchronous transmissions (b) Synchronous transmissions

Figure 1.2: Slot synchronization problem, and reduced collisions thanks to synchronization.

as Reservation-ALOHA [CRW+73], time division multiple access (TDMA), and wireless multihop MAC schemes [CPB+06].

The performance improvement is explained by the reduction of the contention window from $2T$ to T, T being the slot duration, which reduces the packet collision probability. In Figure 1.2 node i transmits according to its internal timing reference τ_i, which indicates the start of transmission. In Figure 1.2(a) references are not aligned, which causes packets to collide, and two packets are lost. In Figure 1.2(b) nodes are slot synchronized, and their packet transmissions are aligned. In this case packet collisions are resolved, and all packets are transmitted successfully.

This simple example illustrates one of the benefits of slot synchronization; throughput is higher in the slotted case. It also illustrates the synchronization problem. In a self-organized manner, how should nodes i, j and k adjust their internal transmission instants in order to align in time?

Scheduling of Sleep Periods

Power consumption is a crucial issue in mobile systems, in particular in networks of embedded devices and sensors. To reduce power consumption, nodes may switch to a power saving mode ("sleep mode") for some time, and the system schedules them to "wake up" simultaneously in order to exchange data. There are various implementations of this concept [DEA06], one of which is the Sensor-MAC (S-MAC) protocol [YHE02]. The basic idea is to compose the slot structure into three parts: a preamble where nodes synchronize their timing, a contention-based communication period where nodes exchange data, and a sleep period where only vital functions are powered. Synchronization is a key issue in S-MAC as nodes must wake up at the same time instant so that communication is not missed. Sleep and wake-up cycles are coordinated in a way that neighboring nodes form virtual clusters to synchronize on sleep schedules. However no synchronization mechanism is put into place for inter-cluster synchronization. Hence nodes that are located near the border of a cluster need to wake up several times within a slot. By establishing inter-cluster synchronization between nodes, all nodes wake up simultaneously, and multiple wake-ups within a slot are avoided. Furthermore the number of transmissions that are needed to form and maintain clusters are reduced, as the whole network performs as a single cluster.

1.3 Outline and Contributions

The remainder of this thesis is composed of five chapters. Chapter 2 defines the network slot synchronization problem, whilst original contributions are presented in the three subsequent chapters. Conclusions are drawn in Chapter 6.

Chapter 2: Synchronization and Networks This chapter gives an overview of synchronization in communication networks. Following this overview, the goals of network synchronization are explained, namely frequency synchronization, time synchronization, and slot synchronization. Each of these forms addresses an imperfection in the clock and can be complementary to the other.

Various decentralized network synchronization techniques have been proposed in recent years, and the most prominent ones are summarized in this chapter. Most of them rely on the exchange of data packets containing local timing information to perform synchronization.

Finally the definition and basic principles of a meshed network are introduced. A meshed network can be characterized by various metrics such as the connectivity level, the node degree distribution, or the minimum node degree. These are detailed and a comparison for meshed networks based on random geometric graphs is given.

Chapter 3: Synchronization of Pulse-Coupled Oscillators The principles behind firefly synchronization are presented and extended in this chapter. The basic synchronization rules are simple, but their analysis is rather complicated. The Pulse-Coupled Oscillator model (PCO model) requires each node to periodically transmit a pulse that indicates the timing of the node, and to adjust the internal timing when receiving an external pulse. These discrete interactions are difficult to treat mathematically and the original proof of Mirollo and Strogatz makes a number of ideal assumptions, i.e. nodes are coupled all-to-all and no interaction delays are present. The convergence analysis of Mirollo and Strogatz is presented, both for two and many nodes, in order to gain insights and intuition into the synchronization process.

As the PCO model forms the basis for performing synchronization in subsequent chapters, a comprehensive analysis under imperfect conditions and some extensions to the model are investigated in this chapter. When coupling delays are introduced in the model, flashing instants are no longer perfectly aligned in time, i.e. the achievable accuracy is no longer zero. The accuracy in meshed networks is extensively detailed, and it is shown that nodes synchronize with an accuracy equal or lower to their coupling delay. Finally this chapter introduces a synchronization metric for meshed networks. This metric quantifies the local level of synchrony, and reveals that the local level of synchrony does not depend on the network size, which is a key advantage of self-organized synchronization.

Notable contributions presented in this chapter include:

- a comprehensive presentation and rewording of the PCO model;
- the evaluation of the impact of delays in meshed networks of PCOs;

- a novel metric quantifying the local synchronization state of PCOs.

Chapter 4: Emergent Slot Synchronization in Wireless Networks This chapter presents and evaluates a novel concept, called Mobile Emergent Firefly Synchronization (MEMFIS). MEMFIS performs network slot synchronization by integrating the synchronization rules of the PCO model into the communication phase, which avoids a periodic re-synchronization phase and the disruption of ongoing data communications. Following this approach, synchronization emerges whilst nodes transmit data. Simulations show that, starting from a random situation, synchronization is reached reliably. Furthermore, as nodes constantly update their clock upon reception of a data packet, clock drifts are compensated, and nodes adapt dynamically to changes in their environment.

MEMFIS relies on the detection of a synchronization word common to all nodes, which is embedded into each packet along with payload data. Its detection is done without any *a priori* knowledge of other nodes' local clocks. This calls for sufficiently long synchronization words that can be reliably detected and distinguished from payload data. The consideration of long synchronization words delays the coupling between nodes, which is considered instantaneous in the PCO model. To combat transmission delays of the physical layer, a delay tolerant approach is taken, so that the achieved accuracy is limited only by propagation delays. Furthermore constraints from a slotted MAC protocol are taken into account, where nodes transmit a synchronization word only when payload data is scheduled for transmission. This is different from the PCO model where nodes transmit pulses at the end of *every* period. In fact the MAC scheduling policy is shown to have a significant impact on the performance.

The performance of MEMFIS is extensively evaluated considering three key requirements of self-organized systems: the scalability toward the network size and traffic density; the adaptability toward different network topologies; and the robustness in case of unreliable detection of synchronization words. MEMFIS is shown to outperform alternative synchronization algorithms where clocks are adjusted based on the weighted average of received synchronization signals [AAK91, SN99, ERLH02]. Simulation results demonstrate that synchronization emerges faster, in particular as the network gets sparser.

Notable contributions presented in this chapter include:

- MEMFIS, a slot synchronization scheme based on the PCO model adapted to wireless networks. MEMFIS relies on the detection of a sync-word common to all that is used to emulate the pulse exchange of PCOs;

- a convergence analysis of MEMFIS by extending the proofs of [MS90, LW04]. MEMFIS achieves an in-phase synchronization regime by delaying the coupling by one period when neglecting propagation delays;

- a derivation of the sync-word detection performance. As all nodes utilize the same sync-word and data is transmitted along, its detection is interfered by these elements;

- the evaluation of MEMFIS thanks to an event-driven system simulator integrating network and data packet generation, coupling interactions, and missed detections and false alarms of sync-words;
- two extensions of MEMFIS to fit predetermined frame structures of current technologies.

Chapter 5: Decentralized Inter-Base Station Synchronization This chapter develops a decentralized inter-base station slot synchronization algorithm suitable for cellular mobile communication systems. The proposed Cellular Firefly Synchronization (CelFSync) algorithm is derived from the theory of pulse-coupled oscillators presented in Chapter 3. In order to maintain synchronization among base stations (BSs), even when there is no direct link between adjacent BSs, some selected user terminals (UTs) participate in the network synchronization process. Synchronization emerges by exchanging two distinct synchronization words, one transmitted by BSs and the other by active UTs, without any *a priori* assumption on the initial timing misalignments of BSs and UTs. In large-scale networks with inter-BS site distances up to a few kilometers, propagation delays severely affect the attainable timing accuracy of CelFSync. To counter this unavoidable effect CelFSync is combined with the timing advance procedure. This procedure is common in current cellular systems, and aligns uplink transmissions of UTs by advancing a scheduled transmission by the estimated propagation delay so that they arrive simultaneously at the BS. By an appropriate combination of CelFSync with the timing advance procedure, a timing accuracy within a fraction of the inter-BS propagation delay is retained.

Notable contributions presented in this chapter include:

- CelFSync, a solution for performing network slot synchronization in a BS-UT-BS manner;
- the integration of the PCO model into a predefined super-frame structure;
- the compensation of propagation delays by combining CelFSync with the timing advance procedure;
- the evaluation in the local-area and wide-area scenarios defined in the European project IST-WINNER, which aims at designing the air-interface of the fourth generation of mobile cellular systems (http://www.ist-winner.org).

Chapter 6: Conclusions and Open Issues Conclusions are drawn in this final chapter, and a summary of the main findings of each chapter is given. Although the work presented in this thesis is self-contained, it is not finite, and some extensions interesting for future studies are presented: both direct extensions complementing the work presented in this thesis, and some indirect extensions deviating from the network synchronization topic.

CHAPTER

2 Synchronization and Networks

The need to align in time and frequency transmitters and receivers, i.e. to synchronize them, is an essential building block in communication systems. Synchronization is present in various forms on many levels of the communication chain. For example, recovering the carrier frequency of the transmitter is necessary to coherently demodulate a received signal. Another form of synchronization requires nodes across the network to agree on a common time reference. Section 2.1 presents the types of synchronization in telecommunication networks from the physical transmission until the application level, where multimedia and data synchronization are present.

This thesis is concerned with *network synchronization*: at the network level, it is useful and sometimes mandatory for communicating entities to synchronize and agree on a common time reference. Clocks are inherently not perfect and thus need to be regularly adjusted to maintain synchronization. Different goals such as synchronization in frequency, in time, and in slot, are achieved by network synchronization. Its definition is elaborated in Section 2.2.

Mutual synchronization enables clocks to align without a central node, in a similar manner to fireflies whose timing is not dictated by a master firefly. Such synchronization techniques are particularly attractive in decentralized networks, and further provide advantages such as the ability to cope with changes in the network topology, resilience against attacks, and other features common to self-organized algorithms. Section 2.3 presents mutual network synchronization solutions for the IEEE 802.11 wireless LAN standard and some extensions to meshed networks, and solutions for wireless sensor networks.

Before presenting the original firefly synchronization model in the following chapter, meshed networks are defined and examined in Section 2.4. Nodes forming a network are usually connected to a subset of neighbors, and are said to form a meshed network. This section presents a common method for modeling such networks, and defines a metric for characterizing the connectivity level of a network.

2.1 Synchronization in Communication Systems

The term synchronization is familiar to many digital communication engineers. For many, synchronization is restricted to acquiring and tracking the transmitter's clock at the receiver, so that synchronous demodulation can be performed. However, this is only one form of synchronization, known as carrier or symbol synchronization. More generally synchronization is important in different areas depending on the level of abstraction and the context. This section summarizes the different forms of synchronization in telecommunications, pointing out the goal of each.

Carrier Synchronization In wireless communication systems, the baseband signal, which contains the modulated information symbols, needs to be up-converted to a higher frequency before being physically transmitted. This higher frequency is termed carrier frequency, and needs to be reconstructed in phase and frequency at the receiver to perform coherent demodulation. This is known as carrier synchronization, and is commonly accomplished by a phase-locked loop (PLL).

Symbol Synchronization A digitally modulated signal is typically composed of a sequence of pulses representing the transmitted symbols. Digital demodulation requires the receiver to perform symbol timing synchronization, so that the received pulses are sampled at the right instant, and information is correctly extracted. Therefore two parameters need to be estimated, namely the symbol timing offset (STO) and the sampling clock frequency offset (SFO). This form of synchronization is sometimes referred to as clock recovery.

Frame Synchronization After information has been extracted from the sequence of pulses or from the subcarriers, frame synchronization is required to delineate successive data frames among the decoded bit stream. Synchronizing on the frame start enables to form bytes and determine their role at different positions in the frames, e.g. determine user channels in Time Division Multiplexing (TDM) systems, or determine the assigned overhead functions such as error check and control information.

Bit Synchronization Bit synchronization is commonly used with two different meanings. The first one refers to symbol synchronization (see above) in the special case of binary symbols (bits). The second meaning is more common and denotes the synchronization of an asynchronous bit stream according to the equipment local clock. This is accomplished by writing the asynchronous source into a buffer at their own arrival rate, and reading them with the frequency of the local clock.

Packet Synchronization In packet-switched networks, the source of information is split into packets that are transmitted or routed independently to their destination. In this case, the receiver needs to equalize the different delays of the received packets, so that it can rebuild the original stream. This packet synchronization is achieved by recovering the original timing from the received

packet sequence through adaptive techniques or by reading the timing information explicitly written in the packet headers (timestamps).

Network Synchronization / Slot Synchronization All above concepts deal with point-to-point synchronization, i.e. the receiver synchronizes to the timing of the transmitter. Another level of synchronization is network synchronization: with regards to the operation of the network, it may be advantageous for all nodes in the network to agree on a common timing, so that all nodes can operate synchronously with the others.

A subclass of network synchronization is slot synchronization. For this particular form of synchronization, time is divided into intervals, denoted slots, and nodes across the network are required to agree on a common reference instant that marks a slot start. This form of synchronization is reminiscent of synchronization among fireflies: each node periodically goes through a slot start or flashing instant, and synchronization is achieved by aligning these reference instants. Network slot synchronization is the main topic addressed in this thesis and its definition is refined in the following section.

Data Synchronization At the user level, data file synchronization is very useful when accessing and modifying data files from different access points. Some examples include keeping an email inbox up-to-date, and accessing the latest version of a file. Typically these data can be modified from different sources and/or by different users, and some process, termed data synchronization, needs to be put into place so that the most up-to-date version of the file is available to the user. Some data synchronization solutions include Subversion [Inc] and FolderShare/Live Sync [Mic], which is integrated into the Windows operating system, for maintaining folders updated, and the email protocol Internet Message Access Protocol (IMAP) [IMA03], which keeps emails on a server and updates the local inbox when a network connection is available.

Multimedia Synchronization Multimedia refers to the integration of heterogeneous elements such as text, images, audio and video in a variety of application environments. These data can be heavily time-dependent, such as audio and video in a movie, and can require time-ordered presentation during use. Multimedia synchronization deals with the alignment in time of these heterogeneous elements, e.g. images, text, audio, video, in a multimedia communication at different levels of integration.

2.2 Network Synchronization

Network synchronization in telecommunications is defined as distributing or aligning time and frequency over a network of clocks located at different locations through the available communication means [LGHD85]. Synchronization is needed to overcome local clock inaccuracies and unavoidable transmission delays.

Network synchronization brings a number of benefits that usually depend on the application. Some examples include:

- interferometry and coordinated multipoint-to-point transmission schemes, a relative time needs to be agreed upon among transmitters so that all transmit synchronously;
- in cellular systems, all base stations need to obey extremely accurate frequency synchronization so that they do not transmit and interfere adjacent frequency bands.
- recording events in a telecommunication network, all nodes should agree on a universal time such as GMT;
- synchronization with regards to the beginning and end of time slots, which is necessary in telecommunications networks which require some form of TDMA scheme, such as satellite networks, GSM mobile terminals, etc.;
- synchronization of clocks located at different multiplexing and switching points in digital telecommunications networks.

These different forms of network synchronization are often categorized according to their goal, i.e. agreement on the frequency for network frequency synchronization, on a common time for network time synchronization, and on the slot start for network slot synchronization. These categories are further clarified in Section 2.2.3 after the model of a clock and the definition of an absolute international synchronization reference are given in Sections 2.2.1 and 2.2.2 respectively.

2.2.1 Modeling and Characterizing a Clock

Each clock is composed of an *oscillator* and a *counter*. The output of a local oscillator u_i describes a periodic signal:

$$u_i(t) = U_i \sin\left(2\pi\Phi_i(t)\right) \ , \tag{2.1}$$

where Φ_i is the total phase of the considered oscillator. The amplitude of the oscillator output U_i is considered constant and is normalized to $U_i = 1$.

The instantaneous frequency is subject to imperfections, which are classified as drift and jitter, and evolves around a nominal frequency f_0 according to:

$$\frac{\mathrm{d}\Phi_i(t)}{\mathrm{d}t} = f_0 + f_\mathrm{d} + \frac{\mathrm{d}f_\mathrm{n}(t)}{\mathrm{d}t} \ . \tag{2.2}$$

The variable f_d models the frequency drift, which refers to long-term deviation between the actual and the ideal clock frequencies. This drift is deterministic, and depends on parameters such as the initial calibration, the external temperature, and the aging of the oscillator. The jitter term $f_\mathrm{n}(t)$ refers to short-term variations in clock frequency; it is known as the phase noise and is a zero-mean random variable characterizing the short-term instabilities.

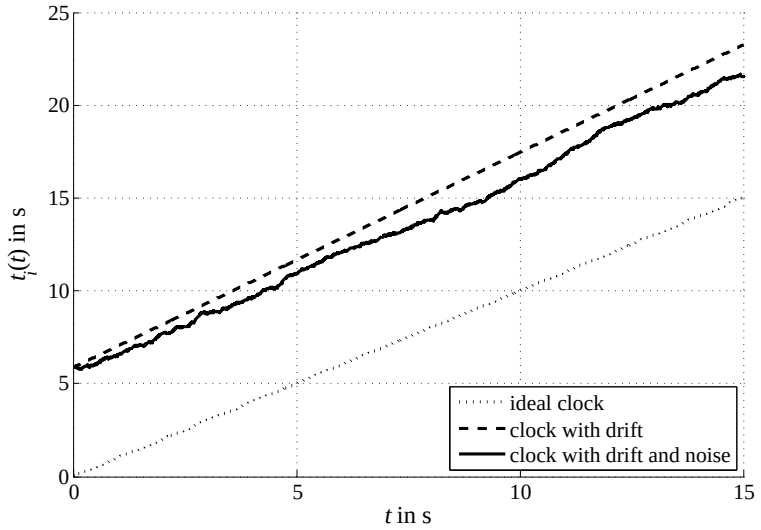


Figure 2.1: Plot of the time process of a clock.

The *total phase* of oscillator i is obtained by integrating the instantaneous frequency (2.2) from 0 to t, which yields:

$$\Phi_i(t) = \Phi_i(0) + f_0 t + f_\mathrm{d} t + f_\mathrm{n}(t) \,, \tag{2.3}$$

where $\Phi_i(0)$ is the initial total phase offset.

Finally, the time process of the considered clock is obtained by dividing the total phase process by the nominal frequency f_0:

$$t_i(t) = \frac{\Phi_i(t)}{f_0} \,. \tag{2.4}$$

The time process $t_i(t)$ completely describes the behavior of a clock, i.e. the counter and the oscillator. It is incremented by the value of the natural period each time the oscillator output $u_i(t)$ finishes one period, and its fractional part relates to the time remaining until the next period.

Figure 2.1 plots an example of a realistic time process following (2.4). The initial phase offset with regards to the ideal time t is equal to 6 seconds. The clock skew increases because the clock drift is strictly positive in this example. Only first order drift is considered, so that the skew increases linearly. Finally phase noise is modeled as Brownian random motion, and causes the time process to vary unpredictably.

In the remainder of this thesis phase noise is neglected for simplicity. This is often the case in network synchronization studies, because the oscillator frequency is relatively low. This assumption further holds by requiring a sufficient quality factor for the local oscillator [Rob84].

Since the fabrication of the first mechanical clock various improvements and clock types have been investigated, and are classified by their ability to keep time. Figure 2.2 summarizes the time-keeping abilities for various clock types.

From this figure the accuracy range of an oscillator decreases over time and depends on the time since synchronization. For example the most accurate oscillator is the superconducting cavity when

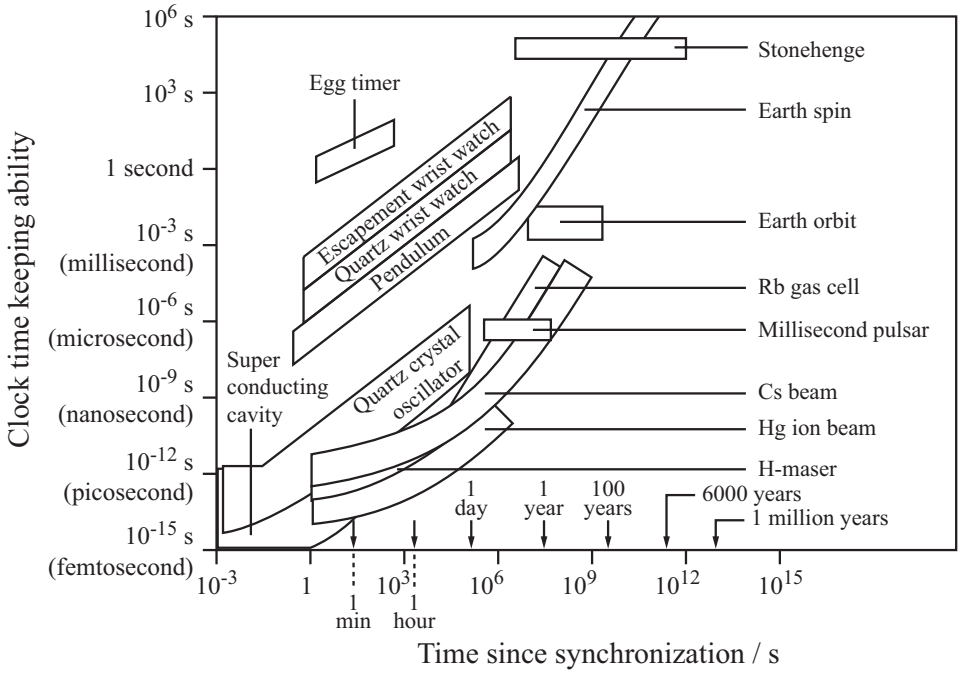


Figure 2.2: Timekeeping ability of different clocks (from [LAK99]).

synchronization is performed every second, but it is outperformed by the Hg ion beam oscillator when synchronization is performed every minute. The performance of other more common and less cumbersome clocks are also shown in this figure, e.g. the caesium beam clock (Cs beam clock), which is often referred to as atomic clock, is accurate up to a microsecond per year, and a pendulum oscillator keeps an accuracy of ten milliseconds per day.

For science and technology it is of great importance to precisely define the duration of a second. In the International System of Units (SI) the definition of a second is one of the seven key units, along with the kilogram, the meter, the ampere, the kelvin, the mole and the candela. Other SI units such as the Hertz are derived from these base units.

The current definition of the second in the international system was adopted in 1967. The Resolution 1 of the 13th General Conference on Weights and Measures (CGPM) defined the second as follows [Bur69]:

> The second is the duration of $9,192,631,770$ periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the caesium 133 (Cs 133) atom.

It follows that the hyperfine splitting in the ground state of the caesium 133 atom is exactly $9,192,631,770$ Hertz. This absolute reference is kept by the International Atomic Time (TAI), which, as a frequency standard, is a weighted average of the time kept by about 200 atomic clocks in over 50 national laboratories worldwide [Bur].

2.2.2 Absolute and Relative Synchronization Reference

Along with the definition of a second, the definition of an absolute origin of time is needed for some applications, e.g. for ordering events in a telecommunication network. The origin is set by a convention, e.g. the Gregorian calendar in the occidental part of the world, which sets the origin of time 2009 years ago at the time of writing.

Internationally the absolute time reference is defined by the Common Universal Time (UTC). As a time scale, UTC divides time into days, hours, minutes and seconds, in the following way:

Sunday 8 March 2009 15:39:25 UTC.

Days are conventionally identified using the Gregorian calendar. Each day contains 24 hours and each hour contains 60 minutes, but the number of seconds in a minute can be 60, or sometimes 61 or 59 on some certain days when leap seconds are needed to align UTC with the Earth's rotation. Thus, in the UTC time scale, the second and all smaller time units such as the millisecond the microsecond are of constant duration, but the minute and all larger time units are of variable duration.

In network synchronization terminology, synchronization to an absolute time reference such as UTC is also known as *time transfer* or *time dissemination*. As the synchronization reference is set by convention, it needs to be transfered from a source that is synchronized to UTC or TAI. This can be achieved, for example, through the Global Positioning System (GPS), which maintains a very accurate synchronization to this reference with the help of four atomic clock onboard each GPS satellite.

In contrast to these absolute time and frequency references, synchronization to a *relative* reference does not require any external tie: nodes in the network agree on a common reference that is valid only within the network, i.e. it is not necessarily valid outside the network. Agreeing on a relative synchronization reference is generally a simpler task than synchronizing to UTC and TAI, and is sufficient for many applications:

- it has been recognized that in wireless sensor networks, the exchange of long timestamps required for UTC consumes much power, and is typically not required [Röm01];
- in digital circuits, synchronous operations are typically determined by the raising edge of the clock. This time reference does not have to relate to UTC for the proper functioning of the circuit;
- in computer systems, the ordering of events among processes running at different locations in a distributed system can accomplished by synchronizing to a "logical" time [Lam78];
- in the manufacturing industry, many machines need to operate synchronously, because the successive steps in the production line need to be performed in a particular order.

Further to the simplicity, relative synchronization provides other advantages such as robustness because no external reference is required, and flexibility in the design of the synchronization network architecture and algorithm.

2.2.3 Categorization of Network Synchronization

Network synchronization can be categorized into *frequency* synchronization, *time* synchronization, and *slot* synchronization. Each category addresses a particular imperfection in the local clock, and typically deals with different time scales. To briefly summarize these categories, frequency synchronization aims at the agreement on the 'tick' intervals among the clocks, time synchronization aims at aligning the counter among clocks, and slot synchronization aims at making clocks tick simultaneously.

Depending on the field of research, the terminology for characterizing synchronization classes may change. Table 2.2.3 clarifies the terminology used in this thesis with regards to terms employed in other fields [Mei05].

In this thesis	Also known as
frequency synchronization	rate synchronization
time synchronization	offset synchronization
slot synchronization	phase synchronization, tick synchronization
relative synchronization	logical synchronization, internal synchronization
absolute synchronization	external synchronization
clock skew	clock offset

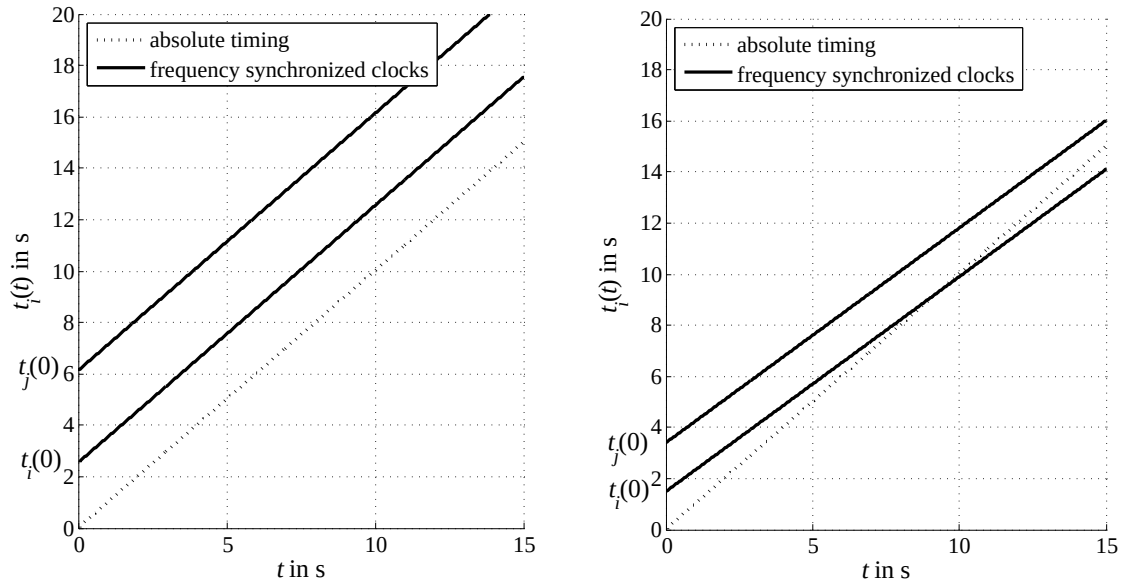
Table 2.1: Synchronization terminology.

Frequency Synchronization

Frequency synchronization in a network is defined as the agreement on the instantaneous frequency among neighboring nodes, i.e. $\mathrm{d}\Phi_i(t)/\mathrm{d}t = \mathrm{d}\Phi_j(t)/\mathrm{d}t, \forall(i,j) \in \mathcal{E}$ where $\mathcal{E}$ is the set of communicating node pairs, but the initial total phase value $\Phi_i(0)$ is not corrected. In other words, clock drift and phase noise in (2.2) are corrected to provide an agreement on a common instantaneous frequency.

Note that network frequency synchronization is sometimes confused with carrier synchronization. Both forms of synchronization address the agreement on a common frequency, but they have distinct goals. Carrier synchronization targets point-to-point synchronization: at the receiver the frequency and phase of the received signal is tracked, so that the carrier frequency of the transmitter is correctly reconstructed (see Section 2.1). Therefore carrier synchronization implies that a master-slave approach is necessary. Network frequency synchronization is more general and involves synchronizing a network: the frequency of local clocks are adjusted, possibly based on information from carrier synchronization, so that remotely located clocks run at the same frequency.

An example of frequency synchronized clocks is shown in Figure 2.3, where noise $f_{\mathrm{n}}(t)$ in the local oscillator is neglected for simplicity. From the definition of frequency synchronized clocks given above, the time process $t_i(t)$ of synchronized clocks have the same slope, but differ with regards to their initial offset $t_i(0)$. Figure 2.3 further illustrates the difference between absolute and relative frequency synchronization. The absolute frequency reference is shown as the perfect clock t,



(a) Absolute synchronization

(b) Relative synchronization

Figure 2.3: Definition of frequency synchronization.

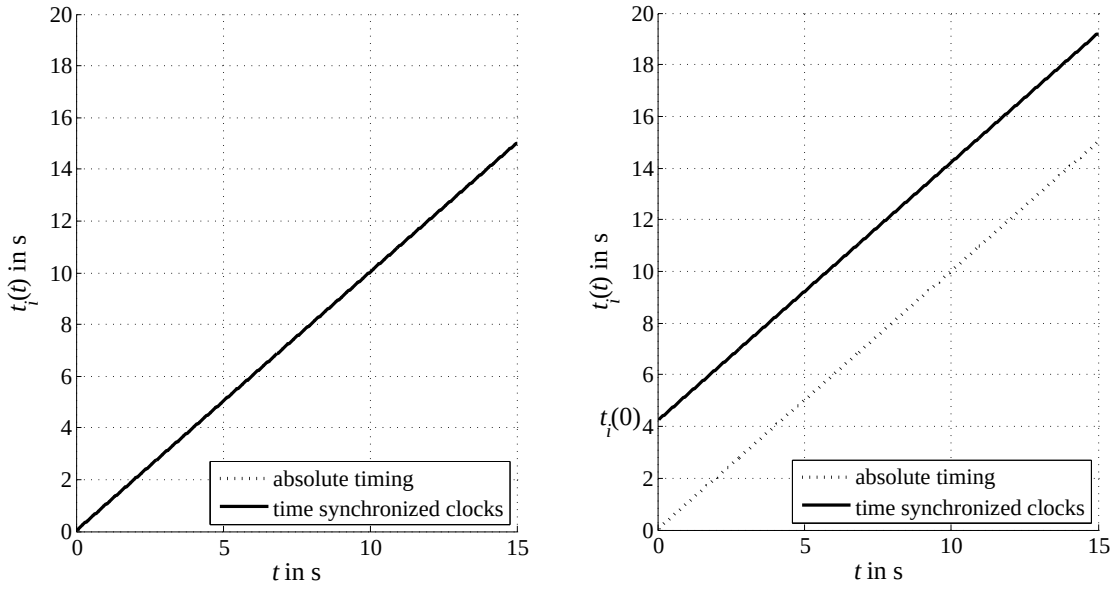
and corresponds to the definition of the second given by the TAI (see Section 2.2.1). For absolute frequency synchronization, all nodes follow the absolute time t so that $\mathrm{d}t_i(t)/\mathrm{d}t = 1, \forall i \in \mathcal{V}$.

Network *frequency* synchronization is an important task required to operate a telecommunication network. For example, in cellular networks, high requirements are put on the accuracy in the frequency of the radio signals to be transmitted. Therefore in the third-generation mobile telecommunications UMTS standard, 3GPP specifies a frequency reference accuracy of 10^{-11} for the core network [3GP04b, 3GP04c] and $5\,10^{-8}$ for the interface between a base station and an attached mobile terminal [3GP04a], which, for a carrier frequency of 2.1 GHz corresponds to a maximum mismatch of 105 Hz.

In wired telecommunication networks, network frequency synchronization is an enabling characteristic for backbone connections. Fiber optic networks standards Synchronous Optical NETwork / Synchronous Digital Hierarchy (SONET/SDH) are organized hierarchically in strata. Each stratum communicates at a given rate, e.g. 51.84 Mbit/s at the lowest level up to 159.3 Gbit/s at the highest, and multiplexing between strata is done in a time division multiplex (TDM) fashion. This requires a high accuracy in the frequency of multiplexers, because data that arrive with a given rate need to be read at the correct pace to avoid missing or overwriting data in the buffer, i.e. to avoid slips [Bre02].

Time Synchronization

Network time synchronization is defined as the agreement on a common time value among clocks in the network, i.e. $t_i(t) = t_j(t), \forall (i,j) \in \mathcal{V}$. This definition is shown in Figure 2.4, where clock



(a) Absolute synchronization

(b) Relative synchronization

Figure 2.4: Definition of time synchronization.

drift and noise are neglected for simplicity. In this figure all nodes agree on a common origin of time $t_i(0)$, and as no further clock imperfection is present, clocks remain synchronized indefinitely.

Absolute time synchronization distributes the time information given by the UTC to the equipment clocks of a network. In telecommunication networks, this form of synchronization is concerned mainly with management purposes. Any relevant event noticed by the equipment monitoring system, such as Bit Error Rate (BER) threshold overflowing, line alarms, hardware failures is recorded for future reporting. The event record includes, among others, the key information date and time as read by the absolute time clock. For this purpose a typical accuracy requirement in the order of a few milliseconds is sufficient [Bre02].

UTC can be accessed through a variety of networks. Examples include navigation systems such as Loran-C [Bla73], and through direct radio broadcasts on longwave radio, e.g. DCF77 (a time signal transmitted every minute from Frankfurt, Germany), and shortwave radio, e.g. WWV (signal transmitted from Fort Collins, USA).

UTC is also distributed by the GPS system. On board each GPS satellite, time is kept by two caesium clocks plus two rubidium clocks, which provide high accuracy and high stability to the time reference. These clocks are updated by ground stations to maintain an accuracy below $1\,\mu$s of UTC. To obtain time from a GPS satellite, a GPS receiver estimates the time difference between the local clock and the GPS clock by combining coarse acquisition signals and data signals transmitted by GPS satellites [PBA02]. Contrary to GPS navigation, which requires signals from at least four satellites, GPS timekeeping requires only one satellite. GPS is a highly accurate time transfer system that achieves a synchronization accuracy of 170 ns [LT91, LAK99]. Furthermore, as GPS

provides direct access to an atomic clock time source, it also solves the problem of absolute network frequency synchronization [Kle96].

If no line of sight to any GPS satellite is available, then GPS synchronization cannot be performed, and other means to obtain access to UTC are necessary. A well-known method to synchronize networks over large distances is the Network Time Protocol (NTP) [Mil91], which was designed to synchronize the Internet. NTP relies on sophisticated mechanisms to access and distribute UTC. NTP servers are distributed in the network, and have direct access to UTC, typically through GPS. Nodes participating to NTP form a hierarchy, where parent nodes provide timing information to leaf nodes. Further sophistications form the protocol, and address the problem of link failure and self-organization into strata. A reported accuracy in the order of a millisecond is achieved thanks to NTP [Mil90, Joh04].

Absolute time synchronization is typically performed based on a centralized approach, which, as pointed out in [Röm01, EGE02], is not well suited for self-organized networks where the network topology changes over time. Furthermore, for applications that require to establish a causal ordering of events, relative time synchronization is sufficient. This form of synchronization is sometimes referred to as logical time synchronization [Lam78]. With the recent increase in research on wireless sensor networks and ad hoc networks, many relative time synchronization techniques have been proposed, some of which are presented in Section 2.3.

Relative time synchronization does not require any external time source, and is valid only among the considered nodes. This form of synchronization has been particularly studied in fault-tolerant systems with the problem known as the Byzantine General problem, which was formulated as follows [LSP82]:

> We imagine that several divisions of the Byzantine army are camped outside an enemy city, each division commanded by its own general. The generals can communicate with one another only by messenger. After observing the enemy, they must decide upon a common plan of action. However, some of the generals may be traitors, trying to prevent the loyal generals from reaching agreement.

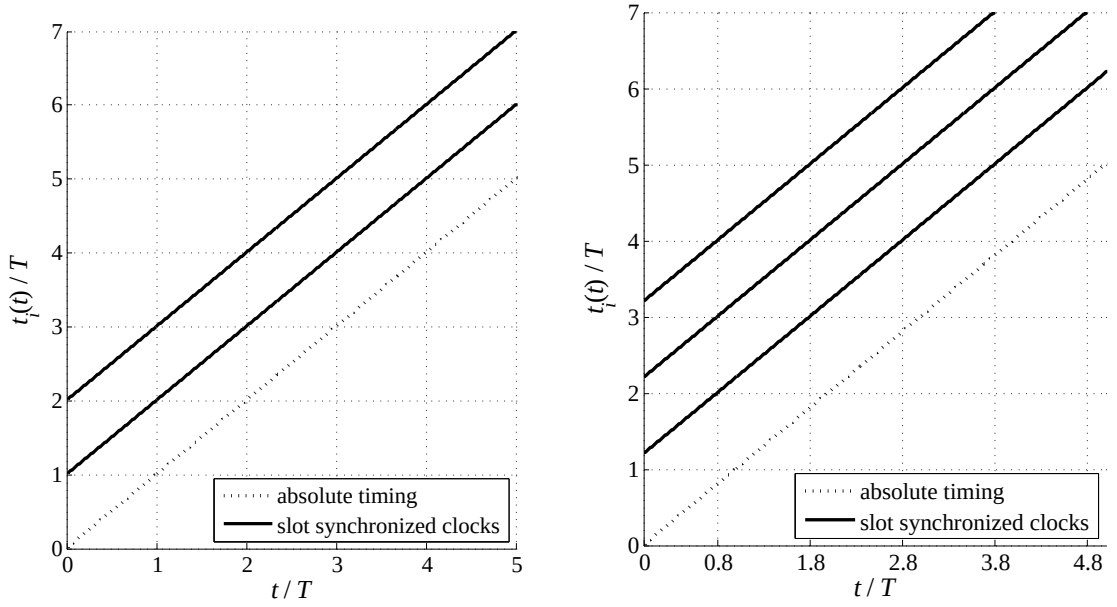
The analogy with relative time synchronization is that clocks exchange their timer information in order to agree on a common one, but some clocks may be faulty or malicious and therefore cause perturbation.

Slot Synchronization

In slotted communication systems, time is divided into periods of fixed duration T, which are known as slots. Slot synchronization is defined as the agreement on a common slot start and end. Some benefits of slot synchronization were presented in Section 1.2.3.

Let $t \in \mathbb{R}_0^+$ denote the absolute time in a system. Consider a set of nodes, each node maintaining an internal clock of natural period T. The slot clock of node i evolves according to:

$$c_i(t) = \exp\left(\mathrm{j}2\pi \frac{t_i(t)}{T}\right), \tag{2.5}$$



(a) Absolute synchronization

(b) Relative synchronization

Figure 2.5: Definition of slot synchronization.

where j is the imaginary number defined by $j^2 = -1$, and $t_i(t)$ is the time process of node i (2.4). Whenever $t_i(t)/T$ yields an integer, the clock is $c_i(t) = 1$. Such a time instant is called *reference instant* of a node and is denoted by $\tau_i[n_i] = n_i T - t_i(t)$, where $n_i \in \mathbb{Z}_0^+$ is the slot number of node i. This reference instant is related to the initial time variable offset by the relation $\tau_i = (t_i(0)/T) \mod (t/T)$.

A system of nodes starts from a situation where the initial time offsets $t_i(0)$ are in general different for each node. Hence, although all clocks run with the same frequency $1/T$, their reference instants $\tau_i[n_i]$ may differ. Interpreting the reference instants as the beginning of a new slot, the objective of slot synchronization is to align the reference instants for all nodes such that $\tau_i[n_i] \bmod T = \tau, \forall i$ where $\tau \in [0, T]$. Then the timing offsets of all nodes converge to an integer multiple of the slot duration. As the objective is to align the reference instants but not necessarily synchronize to the beginning of the same slot number, the term n_i is omitted in the following to simplify the notation.

It is convenient to describe the evolution of a node's slot clock (2.5) in terms of a phase variable $\phi_i(t) \in [0, 1]$ with natural period T, defined by $c_i(t) = \exp(j2\pi\phi_i(t))$. The phase variable is also related to the total phase process $\Phi_i(t)$ in (2.3) as it is the remainder of the modulo operation of $\Phi_i(t)$ by T. The representations $c_i(t)$ and $\phi_i(t)$ are equivalent as they define the same state of a node. In terms of the phase variable, slot synchronization is accomplished when all nodes agree on a common phase, i.e. $\phi_i(t) = \phi(t)$ for all nodes i.

An example of slot synchronized clocks is shown in Figure 2.5 based on the time process of clocks $t_i(t)$. For absolute slot synchronization, the starting phase offset $\Phi_i(0)$ is a integer multiple

of the slot duration T, so that every T seconds, all slot clocks (2.5) are equal to the absolute reference $\exp(\mathrm{j}2\pi t/T)$. Similarly, for relative synchronization, no external tie is required, and slot clocks are equal.

Slot synchronization can be regarded as a special case of time synchronization. If the nominal frequency f_0 of the oscillator is equal to $1/T$, then both types of synchronization try to minimize the time between the clock ticks in the network. However time synchronization requires nodes to further agree on the slot number n_i, whereas slot synchronization requires node to agree only on the next reference instant.

2.3 Technology

The clocks in synchronous networks agree on a common network-wide time, slot and/or frequency reference. Synchronization can be achieved either in a centralized way, i.e. the synchronization reference of a node is dictated by a master node, or in a decentralized way, i.e. nodes communicate with their neighbors in order to agree on a common timing. The former method is straightforward to implement, e.g. when adjusting the time of a wristwatch, the easiest way is to find a nearby clock and copy its time. Decentralized or mutual synchronization is more complex to implement, but, as fireflies in South-East Asia display, it is possible for nodes to synchronize this way. Table 2.3 summarizes both advantages and disadvantages of master-slave and mutual synchronization techniques.

As time transfer is not instantaneous, some delay measurement and compensation is required to estimate the reference clock at the reception instant and adjust the local clock. Furthermore, over time, node loses its synchronization to the absolute or relative time because of clock drift and clock noise, and thus needs to adjust its clock periodically.

As pointed out in [LGHD85] and in Table 2.3, mutual synchronization offers a number of advantages over a centralized approach. For example, the system is robust against changes in the topology, and the achieved accuracy does not decrease with the dimension of the network. On the other hand, the design of self-organized synchronization algorithms is not straightforward, and may lead to instability given the fact that nodes influence each other. The remainder of this section presents some synchronization techniques that aim at mutual slot synchronization.

2.3.1 Wireless LAN IEEE 802.11

The IEEE 802.11 standard enables the formation of wireless Local Area Networks (LAN) [IEE99] composed of stations that are possibly attached to an Access Point (AP). Although telecommunication networks typically rely on centralized methods for performing network synchronization [Bre02], the IEEE 802.11 standard includes both a centralized mode referred to as AP mode, and a decentralized mode, referred to as Independent Basic Service Set (IBSS) mode, which enables to form a network in an ad hoc fashion without the presence of a central AP.

Network synchronization in 802.11 is required for several functions such as:

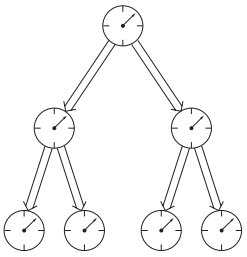
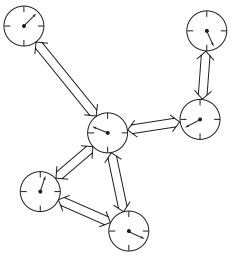
Network structure	Advantages	Disadvantages
Master-Slave / Centralized / Hierarchical 	• No closed loops • Simplicity of implementation	• Network frequency centrally controlled by master • Hierarchical control algorithms related to master failure • Timing error increases with hierarchy level
Mutual / Decentralized / Self-Organized 	• Provides decentralized time and/or frequency control • Lower nodal stability requirements • Provides survival to level of connectivity implemented • Each network node has equivalent influence • Phase and frequency instability of network improves with connectivity implying cheaper nodal clocks/oscillators	• Network frequency depends on path delay dynamics • Required closed loops • Network stability is dependent on path delay dynamics • Relative nodal time errors is path dynamics dependent • Complexity of implementation

Table 2.2: Advantages and disadvantages of centralized and decentralized network synchronization architectures (from [LGHD85]).

- the scheduling of sleep phases: if a station does not have data to transmit or receive, it can enter a sleep period, and wake up regularly to synchronize and receive broadcast information from the AP;

- frequency hopping: nodes need to agree on a common slotted structure, so that they jump to the appropriate frequency at the appropriate moment and obey the appropriate hopping sequence.

The standard requires a synchronization accuracy of 274μs between the nodes in the network [IEE99].

Network Synchronization

The network synchronization function in 802.11 relies on the Timer Synchronization Function (TSF): each node maintains a clock that oscillates with a nominal period of 1μs. The value of the timer is periodically adjusted for synchronization, and depending on the mode of operation, i.e. AP mode or IBSS mode, this is performed in a centralized or decentralized manner.

AP Mode In an 802.11 network with a dedicated infrastructure, the AP is the master and defines the reference time for all dedicated stations. Periodically, data exchange operations are suspended, so that the AP can broadcast its TSF value to the attached terminals. The message containing the TSF information is termed beacon, and the period for beacon transmission depends whether the network is in Frequency Hopping Spread Spectrum (FHSS) mode or in Direct Sequence Spread Spectrum (DSSS) mode. Table 2.3 summarizes the beacon generation parameter, where aCWmin is an integer representing the minimum size of the contention window and aSlotTime is the slot time in μs that the MAC uses for defining the interframe periods. The beacon interval at which the TSF of the AP is transmitted is equal to $(2 \times \text{aCWmin} + 1) \times \text{aSlotTime}$ in both AP and IBSS modes [IEE99]. In AP mode, receiving stations always accept the received TSF and update their clock accordingly.

	FHSS	DSSS
aCWmin	15	31
aSlotTime	50μs	20μs

Table 2.3: Parameters used in generating beacon timer in IEEE 802.11.

IBBS Mode In networks that lack an AP, the TSF synchronization scheme is distributed among all stations. The network synchronization follows a distributed scheme, where beacon frames are generated and sent out in a periodic fashion, in a similar manner to the AP mode. However all stations are allowed to broadcast their TSF information, and receiving stations may adapt or not to this information.

Each station maintains its own TSF timer that is used to schedule a periodic event, where time zero is defined to be a Target Beacon Transmission Time (TBTT). At any TBTT event, a station shall proceed through the following steps [IEE99]:

a) Suspend the decrementing of the backoff timer for any pending non-beacon or non-ad hoc traffic indication transmission.

b) Calculate a random delay that is uniformly distributed in the range between zero and twice aCWmin times aSlotTime (Table 2.3).

c) Wait for the period of the random delay while decrementing the random delay timer.

d) Cancel the remaining random delay and the pending beacon transmission if a beacon arrives before the random delay timer has expired.

e) Send a beacon if the random delay has expired and no beacon has arrived during the delay period.

Upon reception of a beacon, a station does not necessarily update its timer. The received TSF is accepted only if it has a value higher than the local TSF. This avoids clocks going back, which may lead to instability [Lam78].

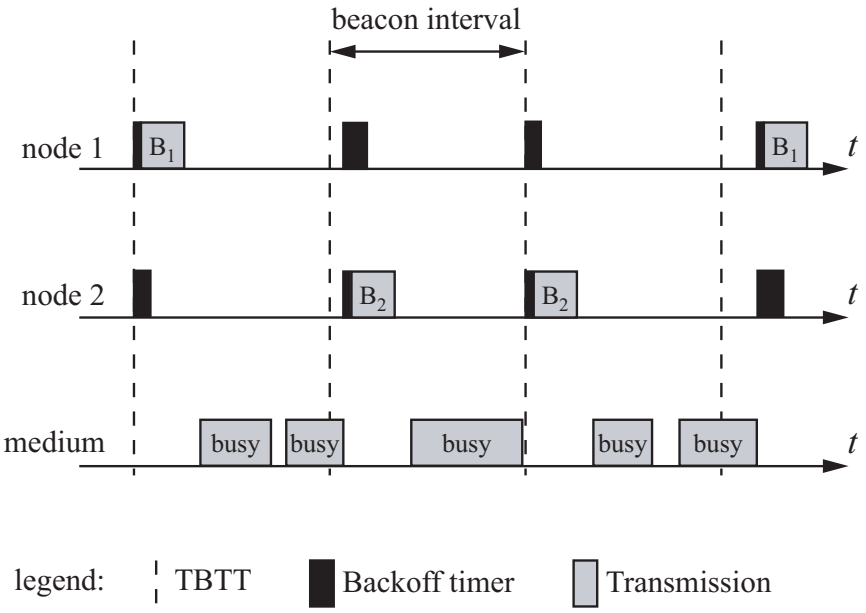


Figure 2.6: Network synchronization principle in the IBSS mode of IEEE 802.11.

The synchronization procedure in IBSS mode is illustrated through an example in Figure 2.6 with two stations denoted node 1 and node 2. In this example, both members compete for beacon generation. At the first TBTT, which corresponds to a reference instant in slot synchronization, the medium is free, and both nodes randomly select a backoff timer. The timer of node 1 expires first, and this node thus transmits its TSF value first, whilst node 2 cancels its timer and receives this information. At the second TBTT, the medium is not yet available, as data is being transmitted. Therefore nodes wait until the medium is again available before suspending data exchange operations, and starting their backoff timer. This time, the timer of node 2 expires first, and its TSF value is broadcast.

Extension to Multihop Networks

The synchronization algorithm described in [IEE99] is targeted at single hop networks, as all terminals need to be able to hear the broadcast information from the AP in order to receive the TSF value.

Extensions to multihop networks, whether several APs need to coordinate and synchronize or stations synchronize over multiple hops, are discussed below. As the first versions of the IEEE 802.11 standard were designed for single hop networks, i.e. all nodes can communicate directly with each other, the performance of TSF synchronization has several issues in multihop networks. In IBSS mode, if the beacon transmitted is not heard by all receiving nodes, then some nodes transmit their beacon, which may disagree with other beacons. Furthermore, in dense networks, the collision probability of beacons is high, and TSF synchronization is sometimes not achieved. These issues are further evaluated in [ZL04].

A variety of schemes have been proposed to overcome the scalability and congestion issues of TSF, so that synchronization is performed for any network topology. The three protocols presented below are fully distributed and scalable. They have similar goals, which are to increase the successful transmission probability for faster nodes and to spread the faster timing information throughout the whole network. The reason behind agreeing on the fastest clock is that nodes should not wind back their clocks when receiving a timestamp, for reasons detailed in [Lam78].

ATSF: Adaptive TSF The Adaptive Timer Synchronization Function (ATSF) was presented in [ZL04]. This scheme modifies the basic 802.11 TSF synchronization by adding a priority scheme. With the basic IBSS synchronization the node with the fastest running oscillator has a higher probability of transmitting its timer information, and is thus a dominant node. To reduce the number of collisions, ATSF implements two counters; a priority counter $I(i)$ is an integer determining the priority of node i with $1 \leq I(i) \leq I_{\max}$, 1 being the highest priority; the second counter $C(i)$ accumulates the number of adopted beacons, i.e. the ones that are not discarded. ATSF updates the priority counter when no beacon is received for $I_{\max}$ consecutive beacon intervals, and it updates the adopted beacons counters when accepting a received beacon.

The performance of the ATSF algorithm was analyzed in [ZL04] and in [RHK04] through simulations. The scalability issue in dense networks is greatly improved with ATSF, while it does not scale well in multihop environment. This is due to the priority counter of ATSF that prevents the periodic generation of beacons and significantly decreases the number of transmitted synchronization messages.

ASP: Automatic Self-time-correcting Procedure The Automatic Self-time-correcting Procedure was presented in [SCS04]. In addition to the basic TSF synchronization protocol, ASP maintains several counters, i.e. the beacon period counter $C(i)$, the period of transmission $P(i)$, and the self-correction interval $A(i)$, and a table for recording the received beacons from neighboring nodes. ASP modifies the basic TSF synchronization for node i by recording received beacons and received timing information, and adjusting the local clock by a self-correction offset that is calculated based on the received timing information.

Similarly to ATSF, ASP nodes with faster TSF transmit more frequently, which enables nodes with slower clocks to progressively join their timing. Simulation results show that an improvement

of 60% over TSF in the average of maximum clock skew. Furthermore, the proportion of unsynchronized clocks, i.e. clocks that are not able to align the common time, is much lower than with TSF.

MTSP: Multihop Time Synchronization Protocol The Multihop Time Synchronization Protocol was presented in [CWH07]. This protocol consists of two phases, a beacon window phase and a synchronization phase. In the former phase, neighboring nodes form synchronization groups, and elect the node with fastest timer as the leader of this group. In the latter phase, leader nodes synchronize with each other.

The algorithm is more complex than ATSF and ASP, and adds overhead to the existing TSF protocol. However it performs better in terms of accuracy and the proportion of synchronized nodes is also higher. Furthermore, the algorithm was evaluated in a dynamic environment, and simulation results showed that clock skew in TSF increases as speed increases, whilst it decreases in MTSP. This is explained by the fact that nodes with faster clocks move and are thus more likely to spread their timing.

2.3.2 Wireless Sensor Networks

Recent years have seen an increase of research efforts on synchronization for Wireless Sensor Networks (WSNs). These networks are not tied to a specific technology, but relate to nodes that are very constrained in energy, resources, and size, and are built for a specific task related to the surveillance of their environment. Synchronization in WSNs is important for several functions [Röm01, EGE02], e.g. to measure the position or speed of objects, to wake up simultaneously and save energy.

TSS: Time-Stamp Synchronization In [Röm01, Röm05], the proposed solution for WSN synchronization does not require nodes to synchronize prior to the measured event to perform synchronous operations. Instead synchronization is performed post-facto by embedding a timestamp along with measurement data, and estimating the time difference between the transmitter's and receiver's clocks and the round trip delay between the two nodes. Afterward the receiving node updates only the timestamp of the received packet (not its local clock) before passing on the message to the destination. With this scheme, a maximum accuracy of 3 ms over 5 hops is achieved [Röm05].

RBS: Reference Broadcast Synchronization With the Reference Broadcast Synchronization algorithm [EGE02], any node can act as a beacon and broadcasts its time reference. Neighboring nodes that receive this reference compare the instant at which they receive the message with other neighboring nodes in order to find the exact broadcast transmission time and to adjust their clock accordingly. Furthermore the algorithm uses a last minute timestamp that is evaluated just before transmission, which reduces random hardware delays and the access time delay, i.e. the time to wait until the channel is free. With this scheme, an accuracy of 3 μs over 1 hop was reported [EGE02].

TPSN: Timing-sync Protocol for Sensor Networks The Timing-sync Protocol for Sensor Networks [GKS03] is based on a hierarchical approach. A tree structure is established amongst nodes in a similar manner to a centralized approach. The round trip delay is measured between parent and child nodes in the tree, and clocks are frequently updated to correct the internal clock. The reported accuracy with this scheme was $17\,\mu$s for a network of 300 nodes [GKS03].

FTSP: Flooding Time Synchronization Protocol In [MKSL04] the Flooding Time Synchronization Protocol was presented, and builds upon the findings of RBS and TPSN. It combines a self-organized algorithm to build a hierarchy with low-level timestamping and clock correction. The top of the hierarchy is elected as master, and periodically floods the network with its timing information. Receiving nodes update their clocks via linear regression of the received timestamps. The protocol was deployed in a network of 60 nodes, and achieved a reported accuracy of $8.64\,\mu$s [MKSL04].

TDP: Time-Diffusion Synchronization Protocol The Time-Diffusion synchronization Protocol [SA05] enables all the sensors in the network to have a local time that is within a small bounded time deviation from the network-wide equilibrium time, which relaxes the tight synchronization constraint usually targeted. The protocol is composed of an active phase and a inactive one, which alternate periodically to maintain synchronization. In the active phase, a hierarchy is first established in a self-organized manner, before the master node starts diffusing its clock value and its clock variance. The achieved accuracy is shown to be bounded by the round trip delay [SA05].

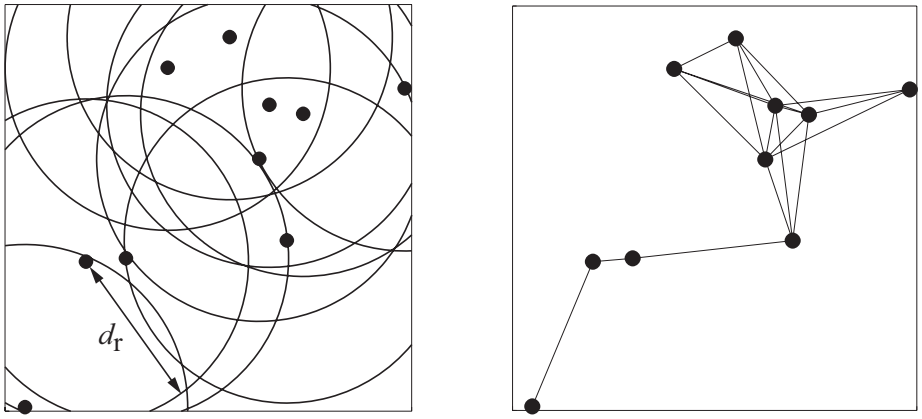
2.4 Meshed Network

Nodes forming a network are usually connected to a subset of neighbors, and are said to form a meshed network. The topology of a meshed network is modeled as a graph $\mathcal{G}$ consisting of a set of N nodes denoted by $\mathcal{V}$ and a set of links denoted by $\mathcal{E}$. Two connected nodes are called *neighbors*. The set of neighbors of node i is defined as $\mathcal{N}_i = \{j : (i,j) \in \mathcal{E}\}$. The number of neighbors of node i, $\left|\mathcal{N}_i\right|$, is the *degree* of the node. If all node pairs are connected by a link, the degree of each node is $N-1$, and the network is said to be fully-meshed.

2.4.1 Random Geometric Graph

A common method to generate random networks is to model the topology as a *random geometric graph* $\mathcal{G}(N, d_\mathrm{r}) = (\mathcal{V}, \mathcal{E})$: N nodes forming the vertex set denoted by $\mathcal{V}$ are placed on a square area of dimension $d_\mathrm{max} \times d_\mathrm{max}$ using a uniform random distribution, and the set of edges $\mathcal{E}$ is formed by node pairs with a maximum distance of d_r.

Figure 2.7 shows an example of a random geometric graph for $N = 10$ nodes. Two representations are shown. In Figure 2.7(a) the maximum communication disc of radius d_r is shown for each node in $\mathcal{V}$. Figure 2.7(b) displays the graph equivalent to this network, where a link is drawn if two nodes are connected. The two representations are equivalent, but the graph representation makes



(a) Communication radius representation (b) Graph representation

Figure 2.7: Example of a random geometric graph of $N = 10$ nodes.

the identification of key parameters such as the number of neighbors easier to grasp. It is therefore preferred.

2.4.2 Network Metrics

Several measures exist to characterize properties of a given network topology. In the following, three measures, namely the algebraic connectivity, the minimum node degree, and the network diameter, are presented, and applied to characterize random geometric graphs.

The *algebraic connectivity* is derived from the Laplacian matrix of a graph, which is itself derived from the degree and adjacency matrix of $\mathcal{G}$. The degree matrix of $\mathcal{G}$ is denoted by $\mathbf{D}(\mathcal{G})$ and is a diagonal matrix of dimension $N \times N$ whose elements are equal to the degree of node i on the i-th diagonal element and 0 elsewhere. The adjacency matrix of $\mathcal{G}$ is denoted by $\mathbf{A}(\mathcal{G})$, and its elements on line i and column j are equal to:

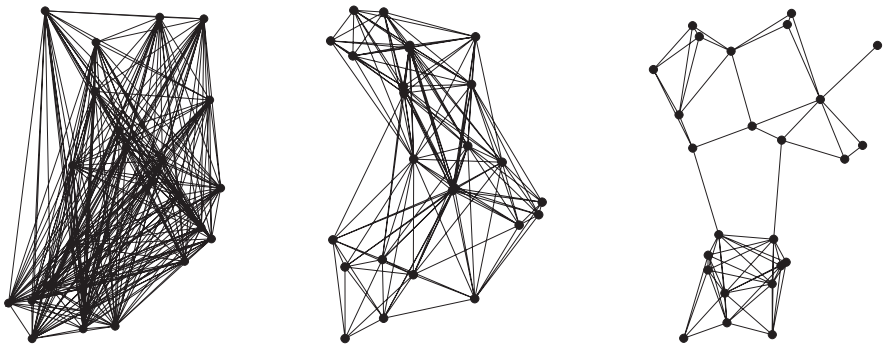
$$a_{ij} = \begin{cases} 1 & \text{if } j \in \mathcal{N}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

For a bidirectional network $a_{ij} = a_{ji}, \forall i, j$.

The Laplacian matrix [Big94] is equal to:

$$\mathbf{L}(\mathcal{G}) = \mathbf{D}(\mathcal{G}) - \mathbf{A}(\mathcal{G}) \,. \tag{2.7}$$

The spectrum of the Laplacian matrix is commonly used to characterize topological properties of $\mathcal{G}$ [Big94]. The second smallest eigenvalue of $\mathbf{L}(\mathcal{G})$ is denoted by κ and is called algebraic connectivity. If the network is connected, i.e. there exists a path between any two nodes, then $\kappa > 0$. If the network is fully-meshed, then $\kappa = N$. The connectivity does not only depend on the number of links in the network, and is an adequate measure of the topology when studying synchronization [MM08]. Figure 2.8 shows three examples of random geometric graphs of 25 nodes for $\kappa/N = 1$, $\kappa/N = 10^{-1}$ and $\kappa/N = 10^{-2}$ respectively.



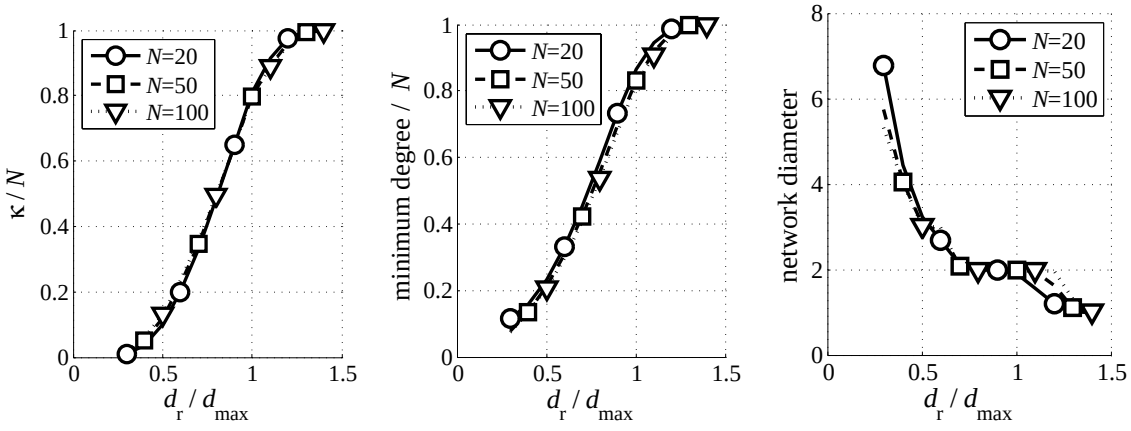
(a) $\kappa/N = 1$ (b) $\kappa/N = 10^{-1}$ (c) $\kappa/N = 10^{-2}$

Figure 2.8: Examples of random geometric graphs with different algebraic connectivities.

Two other common metrics are compared to the algebraic connectivity to further comprehend its behavior:

- the *minimum node degree* is equal to the minimum degree among all nodes in the network. It is therefore the minimum of the diagonal elements of $\mathbf{D}(\mathcal{G})$;
- the *network diameter* is the maximum number of links along the shortest path connecting any two nodes in the network. For example, it is equal to 1 in a fully-meshed network, and it is equal to $N-1$ when nodes form a chain and only have not more than two neighbors.

Figure 2.9 plots the three network metrics against the radius d_{r} normalized by the area side distance $d_{\max}$. These figures are obtained through simulations by generating $5,000$ random geometric networks, which are connected, i.e. there is always a path connecting any node pair, and by varying the maximum distance between connected nodes d_{r}.



(a) Algebraic connectivity (b) Minimum node degree (c) Network diameter

Figure 2.9: Comparison of the algebraic connectivity, the minimum node degree and the network diameter as the radius d_{r} of random geometric graphs changes.

When normalized by the number of nodes N, these three metrics do not change significantly as N varies. Both the algebraic connectivity and the minimum node degree have similar behaviors in random geometric graphs; they monotonically increase as the normalized communication radius d_r/d_{max} augments. Their shape is very similar, but the normalized connectivity varies from 0 to 1 whereas the node degree varies from $1/N$ to 1. On the other hand, the network diameter decreases rapidly as the normalized radius augments, because nodes are brought closer in the sense that fewer hops are needed to connect distant node pairs. When $d_r/d_{max} \geq 1$, the diameter decreases from 2 to 1, i.e. the network becomes fully-meshed.

2.5 Summary

This chapter defined the problem at hand: clocks are distributed at different locations and need to be aligned. For slot synchronization, time is partitioned into segments of equal duration, termed slot, and remote clocks agree on a common slot start. This is in contrast to time synchronization where clocks agree on the same clock value and frequency synchronization that requires nodes to agree on a common measure of time.

Performing network synchronization in a *decentralized* manner brings a number of advantages such as reduced constraints on the deployment of the network, because no central node providing timing information is required. Furthermore a decentralized scheme is robust against node failure and malicious attacks, and the achieved accuracy and stability are improved over centralized schemes (see Table 2.3). Nevertheless a major drawback lies in the difficulty of designing decentralized network synchronization schemes; indeed it is not straightforward to design local update rules that always lead to synchronization. The convergence time, i.e. the time needed to reach a common timing starting from an unsynchronized situation, is another important point.

An existing decentralized synchronization technique is deployed in the wireless LAN standard IEEE 802.11. Synchronization is performed by letting nodes periodically but randomly broadcast their current timer value, and receiving nodes update their clock only if the received timer has a value higher than the received one. This algorithm does not scale well with the node density, and a number of extensions have been proposed to prevent this shortcoming, usually by forming priority levels. A variety of solutions targeted at wireless sensor networks have also been proposed. Similarly to the algorithms for IEEE 802.11, synchronization is performed by exchanging *explicit* timing information.

Finally the theoretical background on meshed networks was presented. This formalism describes networks as a graph related to a set of nodes and a set of links. Random geometric graphs are considered throughout this thesis as the basis for generating random networks. A number of metrics exist to characterize a given network. In particular, for synchronization studies [MM08], the algebraic connectivity seems the most representative and is used in the following; it is derived from the spectrum of the graph and is a scalar that depends on the number of links and on the diameter of the network, among others.

CHAPTER

3 Synchronization of Pulse-Coupled Oscillators

Natural synchronization phenomena are mathematically described as *coupled oscillators*: each entity naturally oscillates and influences others. A striking example is displayed by *fireflies* in South-East Asia: they blink periodically and influence their neighbors through these emissions of light, which, after a transient period, always leads to their synchronization. This chapter details the mathematical model behind firefly synchronization and studies the effect of imperfect conditions on the considered model. The synchronization algorithms presented in Chapters 4 and 5 are built upon the firefly synchronization model and retain most of its characteristics.

The mathematical model behind firefly synchronization is introduced in Section 3.1. Basic notions of a self-sustained oscillator are presented in Section 3.1.1; the oscillator model mathematically describes the internal dynamics of each node. In Section 3.1.2 local rules leading to global synchronization when oscillators are coupled through *discrete* interactions are presented. This model is referred to as pulse-coupled oscillator model (PCO model), and gives simple update rules that are applied when an oscillator perceives a pulse from a neighboring node.

To gain insights into the PCO model, Section 3.2 analyzes the convergence of pulse-coupled oscillators. In spite of the simple rules of the PCO model, the dynamics for many nodes are relatively difficult to treat analytically. The two node case is treated this way in Section 3.2.1, but for many nodes, the main arguments of the formal proof of [MS90] are summarized in Section 3.2.2 and its key features presented.

The synchronization proof for the PCO model assumes that all nodes are coupled to all others. Section 3.3 examines the behavior of pulse-coupled oscillators if nodes form a meshed network (see Section 2.4). As no formal proof is yet available to confirm that synchronization emerges under this condition for any coupling strength, synchronization is investigated through simulations.

Two contributions of this thesis are presented in Sections 3.4 and 3.5; both gain insights into the emerging and steady states of the PCO model.

Section 3.4[1] studies the achieved synchronization accuracy in the presence of coupling delays between PCOs. For a three node network, accuracy bounds in the stable state are derived. This case

[1] Results in this section have been published in parts in [TAB08a] where it received the best student paper award.

study proves useful when looking more generally at meshed networks, where nodes may not be directly connected with all others. While the network topology impacts the achieved accuracy of PCO synchronization, simulations reveal that even for non-neighboring nodes the timing misalignment rarely exceeds twice the direct coupling delay.

Section 3.5[2] presents a local synchronization metric targeted at meshed networks. The proposed metric is computed based on local information only, and enables individuals to *quantify* their local synchronization level. The proposed metric is applied in Section 3.6 to examine the behavior of the PCO model under two imperfect conditions, namely in the presence of oscillator drift and in the presence of coupling delays.

The synchronization rules of the PCO model offer an attractive method to perform decentralized synchronization. It has naturally been applied to different fields in science such as the modeling of neural networks, imaging, and synchronization in communication networks. Section 3.7 summarizes different adaptations of the PCO model studied in the literature.

3.1 The PCO Model

3.1.1 Uncoupled Oscillator

Many systems in Nature, such as a firefly that emits light pulses, a beating human heart, or a pendulum clock, display a common feature: they naturally oscillate. Their rhythm is determined by the properties of the system; an internal source of energy compensates the dissipation. Such oscillators are thus autonomous, and can be described within a class of *nonlinear* models known as self-sustained oscillators.

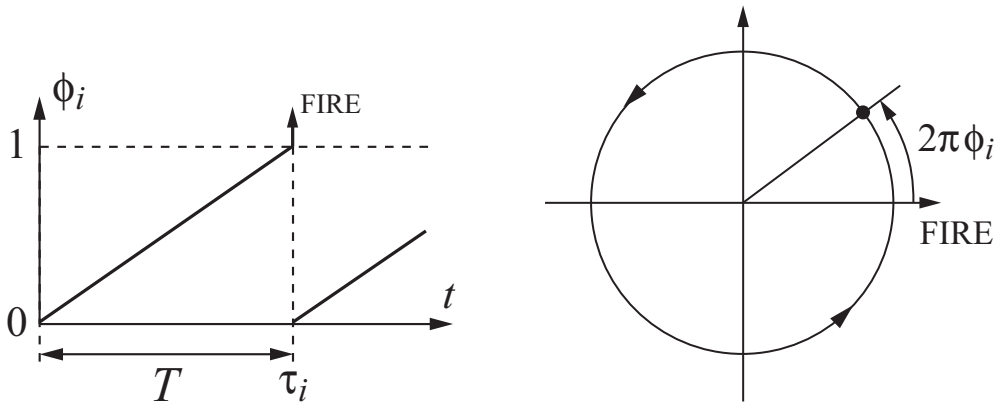
Historically the notion of self-sustained oscillators was first formalized by Andronov and Vitt [AV30]. Prior to this Rayleigh had already distinguished between maintained and forced oscillations, while Poincaré had introduced the notion of the limit cycle; but it was Andronov and Vitt who combined rigorous mathematical methods with physical ideas [PKR01].

Self-sustained oscillators are a subset of the wider class of dynamical systems. This implies that the studied motion is deterministic, i.e. given an initial state, the future state of the system can be unambiguously determined. Dynamical systems are idealized, and do not include noise and other disturbances.

Generally a self-sustained oscillator is represented by a periodic process corresponding, for example, to the angle of a clock's pendulum, the time remaining before a firefly emits light, or the output of an oscillating electrical circuit. In the following a *uniform* oscillator i is considered. Its evolution is entirely characterized by a phase function ϕ_i, which grows linearly over time with a given rate $1/T$:

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}t} = \frac{1}{T} \tag{3.1}$$

[2] Results in this section have been published in parts in [TAB08b].



(a) Temporal evolution (b) Phase portrait

Figure 3.1: Example of the evolution of a uniform oscillator.

where T is the natural period of the considered oscillator. For clarity in the notation, the dependence of the phase ϕ_i over the time variable t is omitted unless necessary.

The phase is normalized so that it is bound to the interval $\phi_i \in [0,1]$. When it reaches the phase threshold normalized to 1, it is said to *fire*: at this instant, denoted *reference instant* τ_i, it resets its value to 0, and in the case of pulse-coupled oscillators, it emits a pulse. Thus, if the phase ϕ_i is undisturbed during one period, it grows from 0 to 1 in T seconds. This evolution is shown in Figure 3.1(a).

Given the definition in (3.1) and the bound to the interval $\phi_i \in [0,1]$, the phase function ϕ_i can be related to the total phase of a clock Φ_i defined in (2.2) by a modulo operation: ϕ_i is equal to the remainder of the division of Φ_i by T. Comparing (3.1) to (2.2) the PCO oscillator model does not incorporate drift and phase noise, because the primary focus of the PCO model is the agreement on a common firing instant when initial phase variables are randomly chosen. The effect of clock drift is investigated in Section 3.6.1.

Clock Based on the phase function, it is convenient to define the internal clock c_i for oscillator i as:

$$c_i = \exp\left(\mathrm{j}\,2\pi\phi_i\right) \ . \tag{3.2}$$

For an uncoupled oscillator following (3.1), the internal clock c_i at a given instant corresponds to the coordinates of the oscillator in the phase plane, and its evolution over time describes the phase portrait shown in Figure 3.1(b); it rotates around the unit circle at constant speed $1/T$.

Phase Portrait The state of an oscillator can be conveniently plotted on a plane, where its evolution over time describes a closed trajectory termed *phase portrait* [Str02]. An example of a phase portrait for a uniform oscillator is shown in Figure 3.1(b). The current phase is indicated by a marker that rotates uniformly over time on the phase portrait. This representation is particularly useful when studying the dynamics of multiple nodes.

3.1.2 Coupling Rules

Oscillators interacting through discrete pulses are said to form a system of *pulse-coupled oscillators*. This model describes systems such as populations of fireflies [BBCH81, MS90], the formation of earthquakes [HH95], and interacting neurons [Izh99]. In [MS90] Mirollo and Strogatz derived a mathematical model for synchronization in populations of PCOs. Under certain coupling conditions, it was proved that, for an arbitrary number of entities and independent of the initial conditions, the network always synchronizes [MS90]. This section details simple update rules leading to in-phase synchronization, i.e. all oscillators firing simultaneously.

When two PCOs are coupled, they are receptive to the pulses of one another, and update their phase function based on the timing of received pulses. Received pulses thus implicitly provide timing information, and a receiving node does not need to know which node has fired.

When node j fires at $t = \tau_j$ and transmits a pulse, node i, which is coupled to node j, instantly increases its phase by an increment $\Delta\phi\,(\phi_i(\tau_j))$:

$$\phi_j(\tau_j) = 1 \Rightarrow \begin{cases} \phi_j(\tau_j^+) = 0 \\ \phi_i(\tau_j^+) = \phi_i(\tau_j) + \Delta\phi\,(\phi_i(\tau_j)) \;\; \forall i \neq j \end{cases} \tag{3.3}$$

where τ_j^+ denotes an infinitesimal time after τ_j.

The phase increment $\Delta\phi\,(\phi_i(\tau_j))$ depends on the current phase of receiving node i and is determined by the Phase Response Curve (PRC). In particular, a simple PRC leading to synchrony is a piecewise linear function [MS90]:

$$\phi_i(\tau_j) + \Delta\phi\,(\phi_i(\tau_j)) = \min\left(\alpha\,\phi_i(\tau_j) + \beta, 1\right) \tag{3.4}$$

where α and β are coupling parameters, which are considered identical for all nodes.

In [MS90], it was shown that, if oscillators are coupled to all others, i.e. a pulse causes a phase increment to all others, and if $\alpha > 1$ and $0 < \beta < 1$, then synchronization is always reached independently of the initial condition, and the time to synchrony is inversely proportional to α, the slope of the PRC.

The PCO synchronization rules are surprisingly simple:

- Each node maintains a phase function according to (3.1) and transmits a pulse when this function reaches a fixed threshold normalized to 1.
- When receiving a pulse, each node increments its phase function by a phase increment $\Delta\phi$ that depends only on the current phase value (3.4).

Interestingly the response of a node in (3.3) depends only on its current phase value as a pulse is received. Thus there is no need to distinguish between transmitters. Furthermore the response of a node to an incoming pulse does not depend on the strength of the received peak. Therefore, if several synchronized nodes transmit simultaneously, their effect on other nodes is identical as the effect of a single pulse. The incidence of a received pulse on the phase function depends solely on coupling parameters, which are constant and the same for all. In Section 3.3 the effect of updating the phase proportionally to the number of simultaneous transmitters is investigated.

3.2 Convergence Analysis

The PCO synchronization rules described in Section 3.1 are applied to a system of two nodes and the process is analytically treated to show that synchronization is always reached. For an arbitrary number of nodes, the proof of [MS90] is summarized and its key features are described.

3.2.1 Two Oscillators

A system of two nodes, characterized by phase functions ϕ_1 and ϕ_2 and evolving according to the PCO synchronization rules (3.3), is considered. Without loss of generality, the system starts after node 1 has fired and is in a state $(\phi_1, \phi_2) = (0, \phi_0)$. This state is shown in Figure 3.2(a). To analyze the dynamics of this system, the system is strobed after each firing instant, and the return map, i.e. the state of the system after each node has fired once, is derived [MS90]. Analyzing the return map enables to show whether the system is always driven to synchrony and to identify any fixed point.

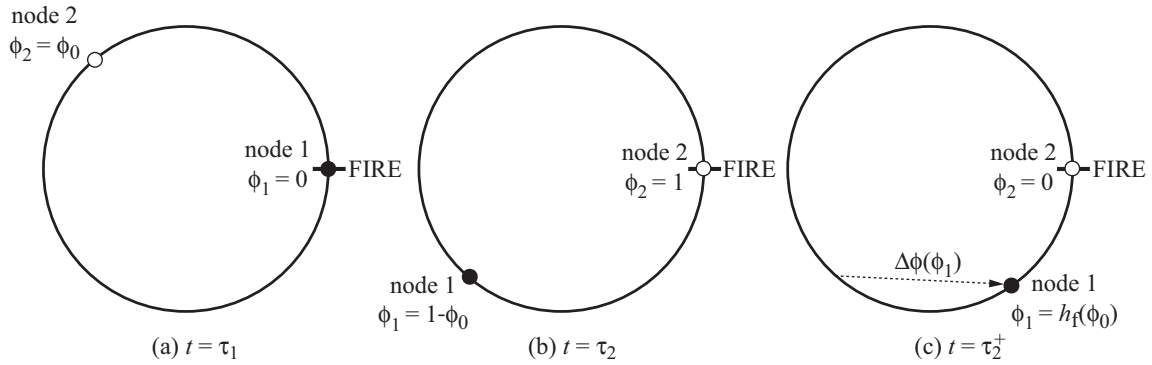


Figure 3.2: System of two oscillators following the PCO rules: (a) after node 1 has fired, (b) when node 2 fires, (c) after node 2 has fired.

Firing Map Nodes interact only through firing events. Therefore, as $\phi_2 > \phi_1$ and nodes have identical dynamics, the firing order cannot change, and node 2 is the next node to fire. From the moment node 1 has fired until the firing of node 2, phases evolve linearly over time according to (3.1) from the initial condition $(\phi_1, \phi_2) = (0, \phi_0)$ to $(\phi_1, \phi_2) = (1 - \phi_0, 1)$ (see Figure 3.2(b)). The firing map h_f is defined as the value of ϕ_1 after receiving the pulse of node 2 and incrementing its phase. This value can be calculated from Figure 3.2(b,c). At this instant, node 2 resets its phase to 0, and the phase of node 1 jumps from $1 - \phi_0$ to:

$$\begin{aligned} h_f(\phi_0) &= \alpha\phi_1 + \beta \\ &= -\alpha\phi_0 + (\alpha + \beta) \ . \end{aligned} \tag{3.5}$$

This equation is valid only if the pulse of node 2 does not force node 1 to fire immediately, which would imply that the nodes are synchronized. This requires that $\phi_1 \in]0, \phi_\ell[$ where ϕ_ℓ is the

absorption limit. This limit is the minimum phase which causes a receiving node to coalesce with the firing node and is derived from (3.4):

$$\phi_\ell = \frac{1-\beta}{\alpha} \,. \tag{3.6}$$

Therefore the firing map (3.5) is valid if $\phi_0 \in]1-\phi_\ell, 1[$.

Return Map After the firing of node 2, the system is in a state $(\phi_1, \phi_2) = (h_\mathrm{f}(\phi_0), 0)$. The next node to fire is node 1, and until this instant, phases evolve linearly until $(\phi_1, \phi_2) = (1, 1-h_\mathrm{f}(\phi_0))$ at $t = \tau_1$,

The *return map* h_R is defined as the phase of node 2 immediately after the *next* firing of node 1, and it is obtained by iterating the firing map one more time:

$$\begin{aligned} h_\mathrm{R}(\phi_0) = h_\mathrm{f}(h_\mathrm{f}(\phi_0)) \quad &= \alpha(1-h_\mathrm{f}(\phi_0)) + \beta \\ &= \alpha^2\phi_0 + (1-\alpha)(\alpha+\beta) \,. \end{aligned} \tag{3.7}$$

This equation is valid if node 2 has not been absorbed by the pulse of node 1, i.e. if $1-h_\mathrm{f}(\phi_0) \in]0, \phi_\ell[$. Equations (3.5) and (3.7) are thus valid if the initial phase of node 2 is in the interval I_ϕ, which is defined as:

$$I_\phi = \left]1-\phi_\ell, 1-\phi_\ell\left(1-\frac{1}{\alpha}\right)\right[\,. \tag{3.8}$$

Figure 3.3 shows examples of the firing and return maps for several values of α and $\beta = 0.01$.

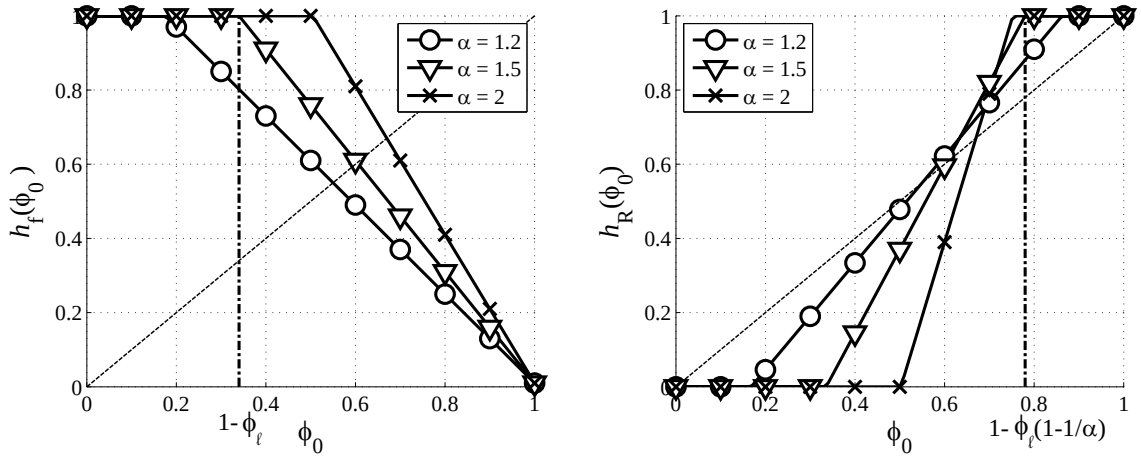


Figure 3.3: Firing map h_f and return map h_R as a function of the initial phase ϕ_0 for several values of α and $\beta = 0.01$. Examples of the boundaries of the interval I_ϕ, i.e. $1-\phi_\ell$ and $1-\phi_\ell(1-1/\alpha)$, are plotted for $\alpha = 1.5$.

Dynamics and Fixed Point After one iteration where each node fires once, the system evolves from an initial state $(\phi_1, \phi_2) = (0, \phi_0)$ to $(\phi_1, \phi_2) = (h_\mathrm{f}(\phi_0), 0)$ after node 2 fires, and to $(\phi_1, \phi_2) = (0, h_\mathrm{R}(\phi_0))$ after node 1 reaches the firing instant again. At this point, nodes have not synchronized if $\phi_0 \in I_\phi$.

Synchronization for the system of two nodes is shown by studying the dynamics of the return map h_R. From (3.7), the slope of the return map is equal to:

$$\frac{\mathrm{d}h_\mathrm{R}(\phi_0)}{\mathrm{d}\phi_0} = 2\alpha\,, \tag{3.9}$$

which is strictly superior to 1 because $\alpha > 1$. Therefore the return map is monotonic strictly increasing in the interval I_ϕ.

Given the dynamics of h_R, it is now necessary to identify any fixed point, i.e. an initial condition $\phi_0 = \phi_\mathrm{fix}$ that does not lead to synchronization, and whether this fixed point is stable or not. By definition of h_R in (3.7), a fixed point for h_f is also a fixed point for h_R, and looking for the fixed points of h_f is thus sufficient. The fixed point equation is defined as:

$$\begin{aligned} F(\phi_0) &= \phi_0 - h_\mathrm{f}(\phi_0) \\ &= (1+\alpha)\,\phi_0 - (\alpha+\beta)\,. \end{aligned} \tag{3.10}$$

Therefore $F(\phi_0)$ describes an affine map, and its slope is equal to:

$$\begin{aligned} \tfrac{\mathrm{d}F(\phi_0)}{\mathrm{d}\phi_0} &= 1 - \tfrac{\mathrm{d}h_\mathrm{f}(\phi_0)}{\mathrm{d}\phi_0} \\ &= 1+\alpha > 0\,. \end{aligned} \tag{3.11}$$

At the limits of the interval I_ϕ, the fixed point equation is equal to $F(1-\phi_\ell) = -\phi_\ell < 0$, and $F(1-\phi_\ell(1-(1/\alpha))) = 1-\phi_\ell > 0$. Thus, as $F(\phi_0)$ is strictly increasing in I_ϕ, there is a unique fixed point ϕ_fix in this interval:

$$\phi_\mathrm{fix} = \frac{\alpha+\beta}{1+\alpha}\,. \tag{3.12}$$

Figure 3.4 summarizes the characteristics of the fixed point equation $F(\phi_0)$. As $F(\phi_0) < \phi_0$ if $\phi_0 < \phi_\mathrm{fix}$, and $F(\phi_0) > \phi_0$ if $\phi_0 > \phi_\mathrm{fix}$, the unique fixed point ϕ_fix is unstable. In summary the stability around the fixed point for the return map is given by:

$$\begin{aligned} h_\mathrm{R}(\phi_0) > \phi_0 \quad &\text{if } \phi_0 > \phi_\mathrm{fix} \\ h_\mathrm{R}(\phi_0) < \phi_0 \quad &\text{if } \phi_0 < \phi_\mathrm{fix} \end{aligned} \tag{3.13}$$

The return map h_R has simple dynamics. After each node has fired once, the distance between their phases is diminished. Iterating several times the return map thus always leads to a synchronous system for any initial condition other than a single unstable fixed point ϕ_fix. The rate of synchronization is controlled by the coupling parameters α and β, which determine the width of the absorption interval (3.6) and thus the coalescence rate.

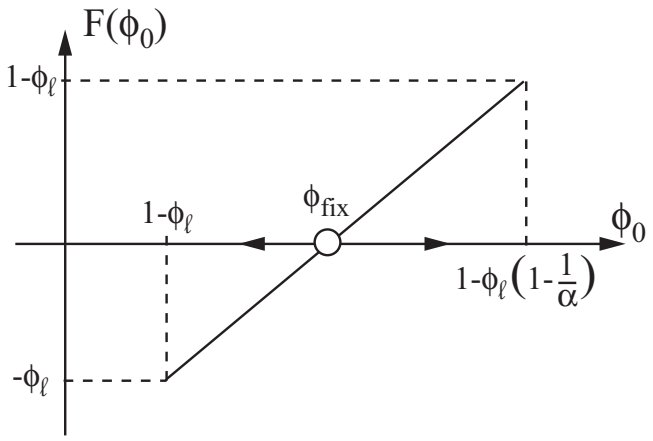


Figure 3.4: Stability of the fixed point for a system of two PCOs.

3.2.2 Population of Oscillators

Synchronization in ensembles of oscillators is a fascinating example of self-organization: oscillators interact and after some time align their internal timing reference regardless of the initial condition. To prove this, the problem is made simpler by assuming that oscillators are identical and are coupled equally strongly to each other. These assumptions were used by Peskin in 1975 to model the synchronization of heart cells [Pes75], and prove the synchronization of two pulse-coupled oscillators in a similar manner to the proof in Section 3.2.1.

Peskin conjectured synchronization in populations of oscillators, and this problem remained unsolved until 1990 when Mirollo and Strogatz published their seminal work [MS90], which showed that, under certain conditions, synchronization *always* emerges in arbitrarily large populations of pulse-coupled oscillators.

Before mathematically detailing the dynamics of the system, the synchronization process is described intuitively. As the system evolves, each oscillator maintains an internal phase function, until one of them reaches the firing threshold. At this instant, nodes interact, and the firing node may bring some nodes to fire immediately. Therefore these nodes coalesce into a single group with the firing node. After some time, as in the two node case, nodes coalesce into a single group and synchronize. Similarly to Section 3.2.1, to formally prove synchronization in a system of N pulse-coupled oscillators, Mirollo and Strogatz used the Poincaré map approach; first the firing map is derived, i.e. the system is strobed when a node fires, and repeating the firing map N times results in the return map. The main theorem in [MS90] shows that the set of initial conditions that never leads to any absorption has a Lebesgue measure of zero.

Dynamics To describe the system, nodes are labeled according to their phase value. This does not cause any loss of generality because this ordering is always preserved when a firing occurs and between these instants. Between firing instants, as the oscillators have identical dynamics, node cannot overtake each other. When a firing occurs, all nodes receive the corresponding pulse, and the

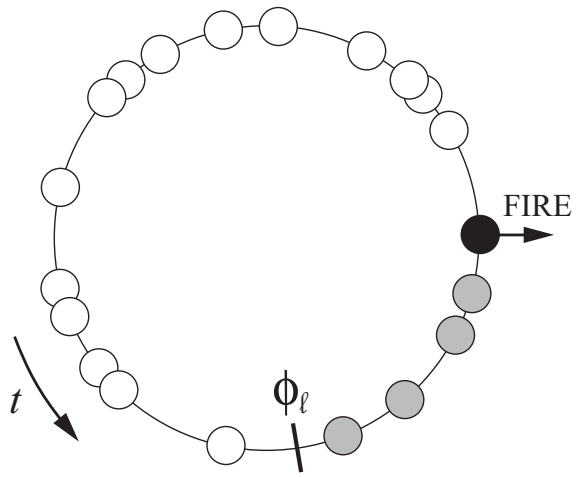


Figure 3.5: Phase portrait representation of $N = 20$ nodes as node N fires.

order does not change, because the phase response curve is strictly monotonic.

With the ordering of nodes, at any given instant, node N is the node that is closest to firing. After node N has fired, the rest of the system is characterized by the remaining phases $\phi_1, \ldots, \phi_{N-1}$, so that the system is described by the set $\mathcal{S}$:

$$\mathcal{S} = \left\{ (\phi_1, \phi_2, \ldots, \phi_{N-1}) \in \mathbb{R}^{N-1} \text{ s.t. } 0 < \phi_1 < \phi_2 < \ldots < \phi_{N-1} < 1 \right\} . \tag{3.14}$$

After firing, nodes are relabeled as follows. Node N is relabeled to node 0, and its phase is reset to $\phi_0 = 0$. All nodes j are then relabeled $j+1$.

Firing Map Similarly to the two node case, the firing map is here derived. The next firing instant occurs when oscillator N fires. Between two firing instants, all phases are increased by $1 - \phi_N$. At this instant, the system is equal to:

$$(\phi_1, \phi_2, \ldots, \phi_{N-1}) = (1 - \phi_N, \ldots, \phi_{N-1} + 1 - \phi_N) , \tag{3.15}$$

and the phase of firing node N is equal to $\phi_N = 1$. At this point, nodes increment their phase by the PRC according to (3.3), and nodes are relabeled as described previously. The firing map is the state of the system at this point.

Absorption The state of the system $\mathcal{S}$ is conveniently represented on a phase portrait common to all nodes. Such a representation is shown in Figure 3.5 for $N = 20$ nodes. In Figure 3.5, node N fires, which causes all other nodes to increment their phase. Given the phase response curve defined previously (3.4), the phase increment is strictly positive. Oscillators thus try to catch up with the firing oscillator by moving closer to the firing instant in the counter-clockwise direction on the phase portrait representation. The system is declared synchronized when all nodes form a single cluster.

According to (3.4) all nodes whose phase is in the interval $[\phi_\ell, 1]$ increment their phase to 1 upon reception of a pulse and coalesce with the firing node. In Figure 3.5, four nodes, marked in gray, merge with the firing node, marked in black, at the next time instant. The set of absorbed nodes is defined as:

$$\mathcal{S}_\ell = \{(\phi_i, \phi_{i+1}, \ldots, \phi_{N-1}) \in \mathcal{S} \text{ s.t. } \phi_i + \Delta\phi(\phi_i) \geq \phi_\ell\} \ . \tag{3.16}$$

This phenomenon is termed *absorption*, and it repeats itself each time a node fires and if nodes are present in the interval $[\phi_\ell, 1]$ (3.16). If nodes are coupled all-to-all, this merger is *permanent* as all nodes oscillate with the same natural period T and are subject to the same interactions.

Synchronization Theorem The main theorem of [MS90], which is not repeated here for simplicity, shows that the set of initial conditions that never leads to an absorption has a Lebesgue measure of zero, in a similar manner to the single fixed point for the system of two nodes. Therefore after some time, some nodes necessarily coalesce with the firing node. Absorptions reduce the dimension of the system $\mathcal{S}$ (3.14), and after a transient period where absorptions occur, the system is always driven to synchrony.

In their proof, Mirollo and Strogatz also show that the coupling strength of a group can be equal to the strength of a single oscillator or to the combined strength of all oscillators, then, in both cases, nodes always synchronize as long as $\alpha > 1$ and $\beta > 0$ in the phase response curve. This feature is investigated in the following section to investigate the impact of coupling nodes proportionally to the number of simultaneously firing nodes in meshed networks.

3.3 Synchronization in Meshed Networks

A strong assumption in the proof of Mirollo and Strogatz [MS90] is that nodes are coupled all-to-all. More generally, a node is coupled to a *subset* of nodes in the network, and the network is said to be *meshed*.

The proof of Mirollo and Strogatz relies heavily on the assumption that pulse-coupled oscillators are coupled all-to-all, because the order of phases cannot change between firings. Nearly 15 years after the seminal work of Mirollo and Strogatz, a formal proof of synchronization in meshed networks was derived by Lucarelli and Wang in [LW04]. This demonstration assumes that the coupling is weak, i.e. $\alpha - 1 \ll 1$ and $\beta \ll 1$, and utilizes results from consensus theory to show that nodes synchronize asymptotically from any initial condition [LW04]. This approach is further elaborated in Section 4.3.

In the following the emergence of synchronization in meshed networks of PCOs without any assumption on the coupling is investigated through simulations. The primary metric used to evaluate synchronization is the time to synchrony, denoted by T_{sync}. It is defined as the time required by the network to synchronize when phases are initially randomly distributed in the interval $[0, 1]$. Two aspects of PCO synchronization are investigated, namely the reception strategy, i.e. increasing

the internal phase proportionally to the number of transmitters and the impact of the algebraic connectivity.

3.3.1 Reception Strategy

As detailed in Section 3.2.2, absorptions are an important aspect of PCO synchronization, and cause some receiving nodes to coalesce with the firing node and join its firing instant. The following simulations compare two receiving strategies. The first one, termed n pulse reception, assumes that upon reception of n simultaneous pulses from a synchronized cluster of neighbors, node i increments its phase proportionally. Iterating (3.4) n times results in a phase increment equal to:

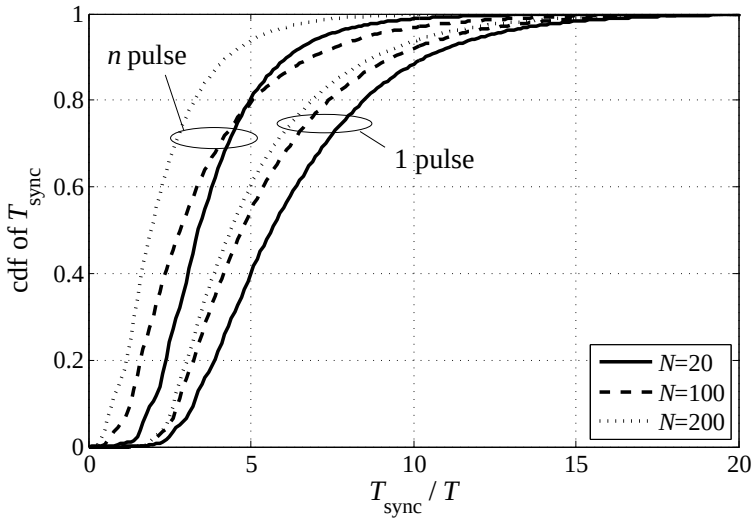
$$\phi_i(\tau_j) + \Delta\phi_n\left(\phi_i(\tau_j)\right) = \min\left(\alpha^n \phi_i(\tau_j) + \beta\left(\sum_{k=1}^{n-1}\alpha^k\right), 1\right). \tag{3.17}$$

This strategy requires nodes to be able to distinguish the number of oscillators firing simultaneously. As all oscillators transmit the same synchronization message, i.e. a pulse, this may not be a simple task. Therefore the second strategy, termed 1 pulse reception, assumes that if several nodes transmit simultaneously, a receiver reacts as if a single pulse is perceived and adjusts its phase according to (3.4).

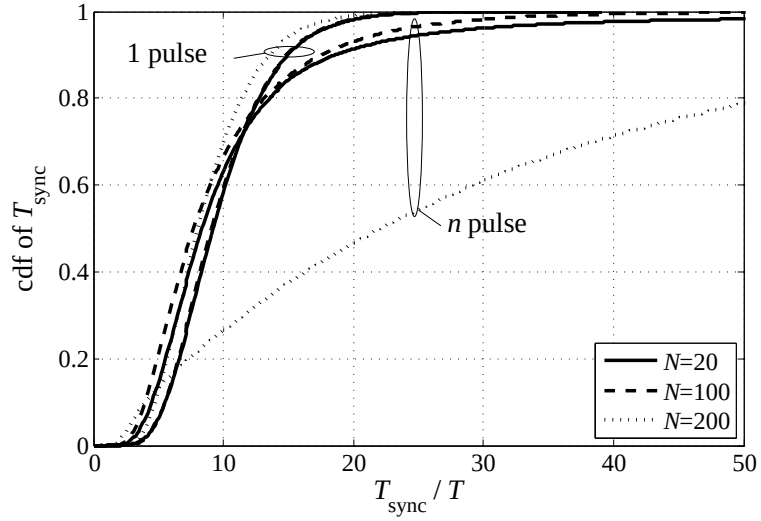
Figure 3.6 compares the n pulse and 1 pulse reception strategies, and displays the cumulative distribution function (cdf) of the time to synchrony T_{sync} based on simulations of 500 random networks of N nodes. For each network, the time to synchrony is measured based on 200 sets of initial phase conditions. The coupling parameters are set to $\alpha = 1.2$ and $\beta = 0.01$.

In Figure 3.6(a), the connectivity is set to $\kappa/N = 10^{-1}$. Both n pulse and 1 pulse reception strategies display similar behavior, and in both cases, synchronization is always reached within 20 periods. The n pulse strategy synchronizes the network faster than 1 pulse: the coupling is increased as clusters of nodes form, such that larger groups are able to gather nodes quickly thanks to the relatively high connectivity of the network and thus absorb more receiving nodes. However the n pulse strategy displays a non-monotonic behavior with regards to the number of nodes N, and the cdf of T_{sync} displays a crossover: for $N = 100$, T_{sync} is lower than for $N = 20$ in 20% of initial conditions, whereas for 1 pulse reception, the time to synchrony always improves as N increases.

The connectivity is lowered to $\kappa/N = 10^{-2}$ in Figure 3.6(b), and a different picture appears. The behavior of 1 pulse reception does not change; all networks synchronize within $25\,T$, and the time to synchrony decreases as the number of nodes increases. On the other hand, with n pulse reception, the performance degrades due to a lower connectivity. For $N = 50$ and $N = 100$, the time to synchrony is lower than 1 pulse reception in 70% of initial conditions. For $N = 200$, T_{sync} is almost always worst with n pulse reception than with 1 pulse, and only 80% of networks synchronize within 50 periods. This performance degradation illustrates the drawback of n pulse reception; this strategy results in a higher coupling and a lower time to synchrony in dense topologies. However in sparser topologies, higher coupling results in instability in the network, in particular if the network size is large ($N = 200$).



(a) $\kappa/N = 10^{-1}$



(b) $\kappa/N = 10^{-2}$

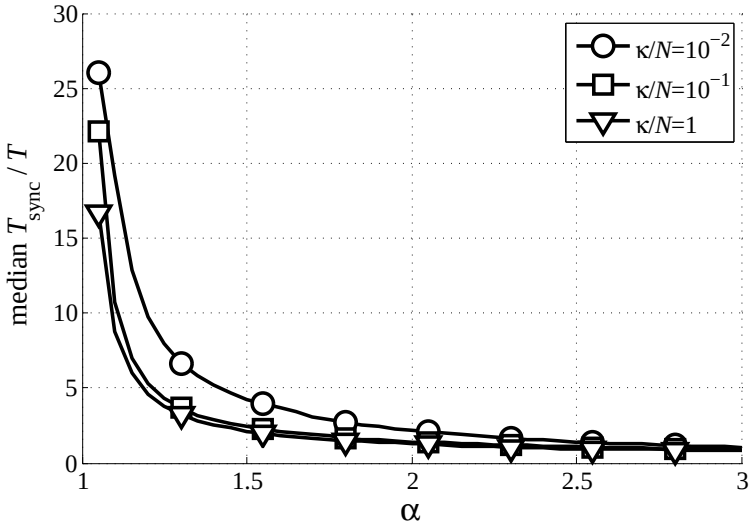
Figure 3.6: Cdf of the time to synchrony in meshed networks of pulse-coupled oscillators as the number of nodes N increases.

Consequently, in the following, only the 1 pulse strategy is considered, as it is more robust to changes in the connectivity and network size. This is further justified by the fact that in wireless networks, estimating the number of transmitters of a common synchronization word is an extra task that is not straightforward.

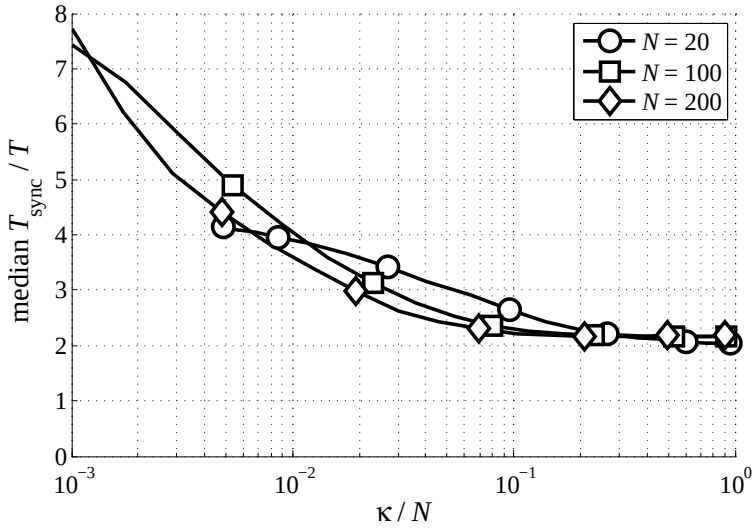
3.3.2 Influence of the Network Connectivity

As observed in Figure 3.6, the algebraic connectivity plays an important role in the time to synchrony [TWG04]. Figure 3.7 examines the impact of the coupling parameter α and of the network connectivity κ on PCO synchronization. In Figure 3.7(a) the number of nodes is set to $N = 50$ nodes and the coupling parameter α varies whilst β is set to $\beta = 0.01$. In Figure 3.7(b) coupling parameters are set to $\alpha = 1.5$ and $\beta = 0.01$.

In Figure 3.7(a) the median time to synchrony decreases inversely proportional to α. This feature was shown by Lucarelli and Wang for weak coupling [LW04] and is here verified for any coupling



(a) Influence of coupling parameter α



(b) Influence of connectivity κ

Figure 3.7: Median time to synchrony in meshed networks of pulse-coupled oscillators.

strength.

In Figure 3.7(b) the median time to synchrony is shown for three values of N as the algebraic connectivity increases. In this case as well, T_{sync} decreases inversely proportional to κ, i.e. nodes synchronize faster when the network is dense. The number of nodes N has little impact on T_{sync}.

3.4 Synchronization Accuracy

A critical assumption in the PCO model presented in Section 3.1 is that transmitted pulses *instantly* influence the phase of receiving nodes. In many systems however delays are unavoidable. For example, when applying the PCO model to wireless networks, propagation delays between nodes, i.e. the time required for a pulse to propagate from the transmitting to the receiving node, need to be taken into account. Delays impact the stability of the PCO model, which needs to be appropriately modified so that stability is regained [MM96]. Although nodes regain stability they can no longer synchronize perfectly, and the accuracy in the stable state is compromised. This section looks at the stable states achieved by systems of two and three nodes in the presence of delays. These simple cases are then used to understand the behavior in meshed networks.

3.4.1 Two Nodes

Consider two PCOs i and j whose coupling is delayed by ν_{ij}. When delays are introduced, such as propagation delays, a system of PCOs may become unstable, and the system is unable to synchronize [EPG95a].

Stability Condition To regain stability, a refractory period of duration T_{refr} after transmitting is introduced [MM96]. In refractory, i.e. when $\phi_i < \phi_{\text{refr}}$ with $\phi_{\text{refr}} = T_{\text{refr}}/T$, no phase increment is possible, so that received pulses are not acknowledged. The refractory period is a common feature in self-organized biological systems and is important for stability [Joh01]. It is present in neurons cells, which require some time to resume to their stable state after firing and do not acknowledge any received pulse during this time.

The duration of the refractory period needs to be at least twice the propagation delay, so that *echoes* are not acknowledged: if node i fires at τ_i and forces node j to fire at $\tau_j = \tau_i + \nu_{ij}$, then the echo transmitted by node j is not acknowledged if node i is in refractory at $\tau_i + 2\nu_{ij}$. The stability condition for this system is [MM96]:

$$T_{\text{refr}} > 2\nu_{ij} \; . \tag{3.18}$$

Under this condition, the system achieves a synchronization limited by the delay between the two nodes [MM96]:

$$\tau_i \in [\tau_j - \nu_{ij}, \tau_j + \nu_{ij}] \; . \tag{3.19}$$

To understand this limitation, the interval of firing instants leading to a stable state are detailed. Nodes i and j are initially synchronized such that node i imposes its delayed reference on node j by

forcing it to fire at $\tau_i + \nu_{ij}$ (in a similar way to nodes 1 and 2 in Figure 3.9). If the reference instant from node j is higher than $\tau_i + \nu_{ij}$ (τ_2 shifted to the right in Figure 3.9), τ_j is reset to the delayed timing $\tau_i + \nu_{ij}$ as long as $\phi_j(\tau_i + \nu_{ij})$ is in the absorption interval at this instant.

On the other hand, if the reference instant from node j is lower than $\tau_i + \nu_{ij}$ (τ_2 shifted to the left in Figure 3.9), the pulse from node i falls into the refractory period of node j, and is thus not acknowledged. This system remains in a stable state as long as $\tau_j \geq \tau_i - \nu_{ij}$, because both transmitted pulses fall within refractory. If $\tau_j < \tau_i - \nu_{ij}$, the roles are reversed, and node j imposes its delayed timing onto node i, i.e. $\tau_i = \tau_j + \nu_{ij}$.

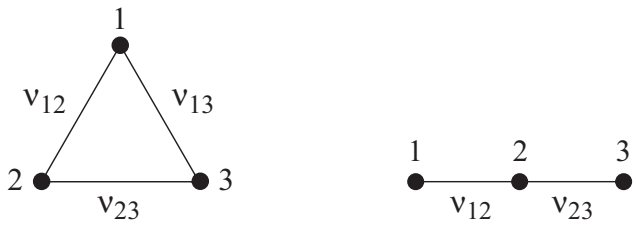
Accuracy The synchronization accuracy between two nodes i and j is defined as the absolute difference in their reference instants:

$$\varepsilon_{ij} = \left|\tau_i - \tau_j\right| . \tag{3.20}$$

Condition (3.19) shows that, in a stable state, the accuracy between directly connected nodes is not larger than the delay, i.e. $\varepsilon_{ij} \in [0, \nu_{ij}]$.

3.4.2 Three Nodes

The three node case is here studied. For $N = 3$, only two connected topologies are possible, and are shown in Figure 3.8.



(a) Triangle topology (b) Line topology

Figure 3.8: Meshed networks of three nodes.

Triangle Topology In a triangle network (Figure 3.8(a)), all nodes can communicate directly with all others. As in the two node case, the system remains stable if no echoes are received. This condition is met if the refractory duration, taken to be the same for all nodes, satisfies (3.18).

Starting from a random initial condition, the system converges to a stable state where condition (3.19) is met for all connected node pairs. A possible state is node 1 firing first and imposing its *delayed* timing onto the other nodes. This state is shown in Figure 3.9: node 1 fires at instant $t = \tau_1$, which causes nodes 2 and 3 to increment their phases at instants $\tau_1 + \nu_{12}$ and $\tau_1 + \nu_{13}$ respectively. Assuming that their phase is higher than the absorption threshold, then nodes 2 and 3 fire at these instants, i.e. $\tau_2 = \tau_1 + \nu_{12}$ and $\tau_3 = \tau_1 + \nu_{13}$, and enter refractory. No further phase increment

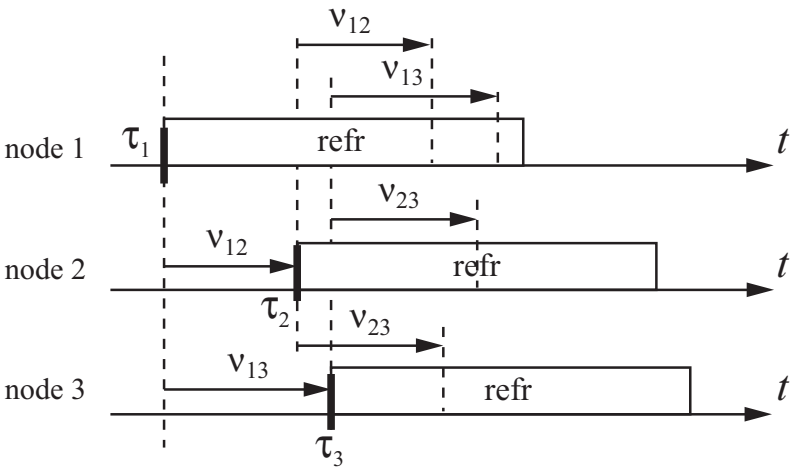


Figure 3.9: Synchronization interactions for three nodes connected all-to-all.

occurs within this period, because the pulses from nodes 2 and 3 are received when nodes are in refractory (3.18). The system thus remains in this stable state.

In this example, the achieved accuracies between nodes 1 and 2, as well as 1 and 3 are exactly equal to the delays ν_{12} and ν_{13}. Interestingly, the accuracy between nodes 2 and 3 is equal to the difference in delays with the forcing node, i.e. $\varepsilon_{23} = |\nu_{12} - \nu_{13}|$. Thus the achieved accuracy does not depend on the direct delay ν_{23} but on the delay difference with the forcing node, and nodes 2 and 3 do not influence each other in this state (see the two coupling instants delayed by ν_{23} in Figure 3.9).

Line Topology When nodes form a line network (Figure 3.8(b)), nodes 1 and 3 cannot communicate directly. Condition (3.19) leads to an accuracy interval after two hops equal to $\varepsilon_{13} \in [0, \nu_{12} + \nu_{23}]$.

Two stable states where one node forces its delayed timing on others are distinguished. If node 2 fires first and entrains the others, it imposes its delayed timing onto the edge nodes 1 and 3. This results in a state similar to the one shown in Figure 3.9, and, although edge nodes cannot communicate directly, the achieved accuracy between them is equal to $\varepsilon_{13} = |\nu_{12} - \nu_{13}|$. If either node 1 or 3 fires first and imposes its timing onto node 2, which then imposes its timing onto the other edge node, the accuracy between the edge nodes 1 and 3 is equal to the sum of propagation delays $\varepsilon_{13} = \nu_{12} + \nu_{23}$.

3.4.3 Many Nodes

The achieved accuracy is extended to larger networks. Determining the achieved stable state of a system of PCOs with delays and of arbitrary size is a difficult task. In a fully-meshed network where all nodes are coupled directly to all others, the two node case can be generalized as follows: if the refractory period is at least equal to twice the maximum delay, the first firing node forces its

delayed firing instant onto other nodes, because it discards their echoes. This problem is treated through simulations, and the simple cases presented for two and three nodes are used to understand the achieved accuracy.

Scatter Plot

To evaluate the stable synchronized state, the achieved accuracy is measured after nodes have synchronized. Figure 3.10 plots a scatter plot of the achieved accuracy ε_{ij} as a function of the propagation delay ν_{ij}: each point on the plot correspond to the accuracy for each node pair (i, j) as a function of their direct propagation delay. Results in this figure are obtained for a meshed network of $N = 25$ nodes with normalized algebraic connectivity $\kappa/N = 10^{-2}$, and differentiate between neighboring links, i.e. $(i, j) \in \mathcal{E}$, and nodes that are not able to communicate directly. Both propagation delays and achieved accuracies are normalized by $\nu_{\max}$ the maximum delay between two connected nodes, i.e. $\nu_{\max} = d_{\mathrm{r}}/\mathrm{c}$ where d_{r} is the maximum distance between two connected nodes and c is the speed of light.

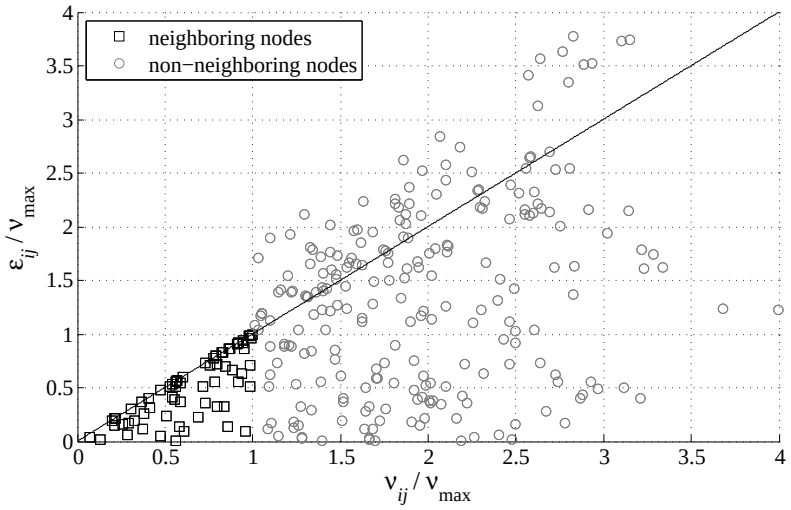


Figure 3.10: Scatter plot of the achieved accuracy ε_{ij} as a function of the propagation delay ν_{ij}.

Results in Figure 3.10 confirm (3.19): the achieved accuracy between neighboring nodes is always equal or lower than the propagation delay, i.e. $\varepsilon_{ij}/\nu_{\max} \leq 1$ for $(i, j) \in \mathcal{E}$. For non-neighboring nodes, the accuracy is more scattered and sometimes exceeds the direct propagation delay. However it is sometimes very close to zero, even for large propagation delays, which confirms what was observed for three nodes (see Figure 3.9).

Normalized Accuracy

To evaluate how often the accuracy is lower than the propagation delay, the distribution of the achieved synchronization accuracy is examined. To this end, it is useful to define ρ_{ij}, the achieved

accuracy normalized by the propagation delay:

$$\rho_{ij} = \frac{\varepsilon_{ij}}{\nu_{ij}} \,. \tag{3.21}$$

Local and Global Accuracies Figure 3.11 plots the cumulative distribution function of the normalized achieved accuracy ρ_{ij} for networks with a normalized algebraic connectivity of $\kappa/N = 10^{-2}$ (see Figure 2.8(c)). The cdf is computed from 200 sets of initial conditions performed on 50 randomly generated networks of $N = 25$ nodes.

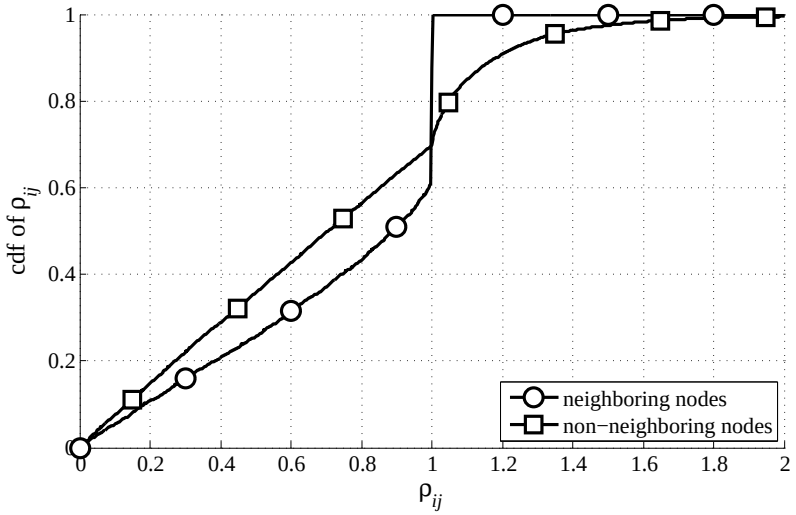


Figure 3.11: Cdf of the achieved accuracy for networks of 25 nodes and an algebraic connectivity of $\kappa/N = 10^{-2}$.

Figure 3.11 confirms results of the scatter plot. The synchronization accuracy among neighbors is below or equal to the propagation delay, i.e. $\rho_{ij} \leq 1$ in all cases. The cdf of ρ_{ij} also shows that 40% of neighboring links have an accuracy exactly equal to the propagation delay. In 70% of cases, non-neighboring nodes achieve an accuracy lower than the direct propagation delay. Among these nodes, the achieved accuracy is always lower than twice the propagation delay, i.e. $\rho_{ij} < 2$ in all cases.

Influence of the Number of Nodes Figure 3.12 plots the achieved normalized accuracy as the number of nodes N varies. The normalized algebraic connectivity is constant and equal to $\kappa/N = 10^{-1}$, and no distinction is made between neighboring and non-neighboring nodes.

Figure 3.12 indicates that as the number of nodes in the network increases, the normalized accuracy improves, and for networks of over 100 nodes, $\rho_{ij} \leq 1$ in all cases. This indicates that middle nodes imposing their timing, as for the three node case forming a line topology, is more present as N increases, which results in lower accuracy. The abrupt jump at $\varepsilon_{ij} = 1$ is still observed, and the height of the jump is proportional to the number of links where neighboring nodes have an accuracy exactly equal to the propagation delay with their neighbors.

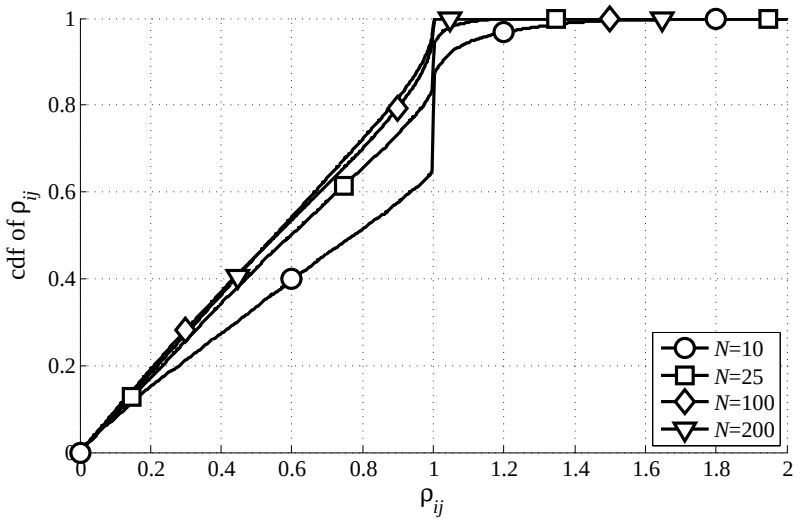


Figure 3.12: Cdf of the normalized accuracy for networks of algebraic connectivity $\kappa/N = 10^{-1}$.

Influence of the Connectivity Figure 3.13 plots the normalized accuracy for different algebraic connectivities and for a constant number of node $N = 50$.

In Figure 3.13, for the fully-meshed case, i.e. $\kappa/N = 1$, the normalized accuracy is always equal or below 1. This confirms the results that the achieved accuracy in a fully-meshed network is never larger than the propagation delay. As fewer direct links are present in the network and the connectivity diminishes, the accuracy also decreases; the jump at $\rho_{ij} = 1$ diminishes as κ decreases, because the size of the set of neighbors is smaller. For a very low connectivity $\kappa/N = 10^{-3}$, the accuracy is no longer always inferior to 2. With such a low connectivity the network topology

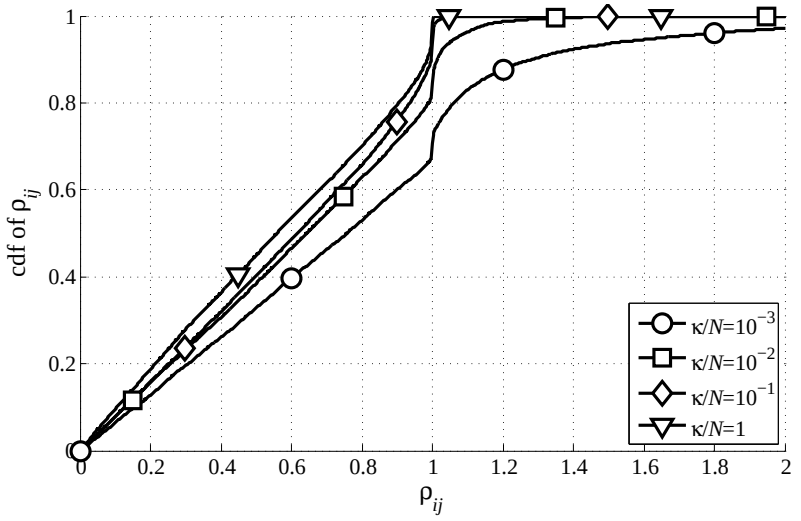


Figure 3.13: Cdf of the normalized accuracy for networks of $N = 50$ nodes with various algebraic connectivities κ.

resembles a line topology, which implies that the delays add up in a similar fashion to the three node case presented in Section 3.4.2.

3.5 Measuring Synchrony

So far, the studies of the PCO model presented in Sections 3.3 and 3.4 examined separately the transient phase where nodes start from a random initial condition (measured by the time to synchrony) and the stable state where, unless a perturbation occurs, nodes fire periodically every T seconds. To bridge these two phases and determine the synchronization state of the considered system, it is necessary to define a synchronization metric. Its objective is to identify whether a network is synchronized and to quantify the state of the network. When all nodes are synchronized in-phase, i.e. they all fire at the same instant, the metric should be equal to 1. When the system is in disorder, i.e. firing instants are randomly distributed within $[0,T]$, the metric should approach 0.

This section details a global metric derived by Kuramoto [Kur84]. Then a new metric targeted at meshed networks of coupled oscillators is presented, and applied to two particular network topologies to motivate its behavior.

3.5.1 Global Synchronization

In [Kur84] Kuramoto studied a simple model of N coupled oscillators interacting continuously, rather than through discrete pulses as PCOs. Each oscillator is described by a phase function as in (3.1), but the phase evolution is continuously updated by the sum of sines of the phase difference between the considered node and all others [Kur84].

To analyze the system of oscillators, the *mean field* of phases is introduced, defined as:

$$\begin{aligned} r\,\exp(\mathrm{j}\,2\pi\phi) &= \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\exp(\mathrm{j}\,2\pi\phi_i) \\ &= \tfrac{1}{N}\textstyle\sum_{i=1}^{N} c_i \end{aligned} \tag{3.22}$$

where r is the Kuramoto synchronization index, and $\phi\in[0,1]$ is the mean phase of all N oscillators.

The mean field is an indicator of the coherence due to synchronization in the network. If all phases are equal, the complex exponentials in (3.22) add up constructively and $r=1$. If phases are uniformly distributed in $[0,1]$, then $r\to 0$. Thus r is an appropriate metric for estimating synchronization in the whole network.

In particular, the Kuramoto metric for $N=2$ nodes evolves according to:

$$r=\sqrt{\tfrac{1}{2}\big(1-\cos\left(2\pi(\phi_j-\phi_i)\right)\big)}\,. \tag{3.23}$$

The Kuramoto metric is not specific to any oscillator model, and can be applied to the PCO model presented in Section 3.1.2. As an example, Figure 3.14 plots the evolution of $N=20$ phases following the simple rules of (3.1) and (3.3), and the corresponding Kuramoto synchronization index r.

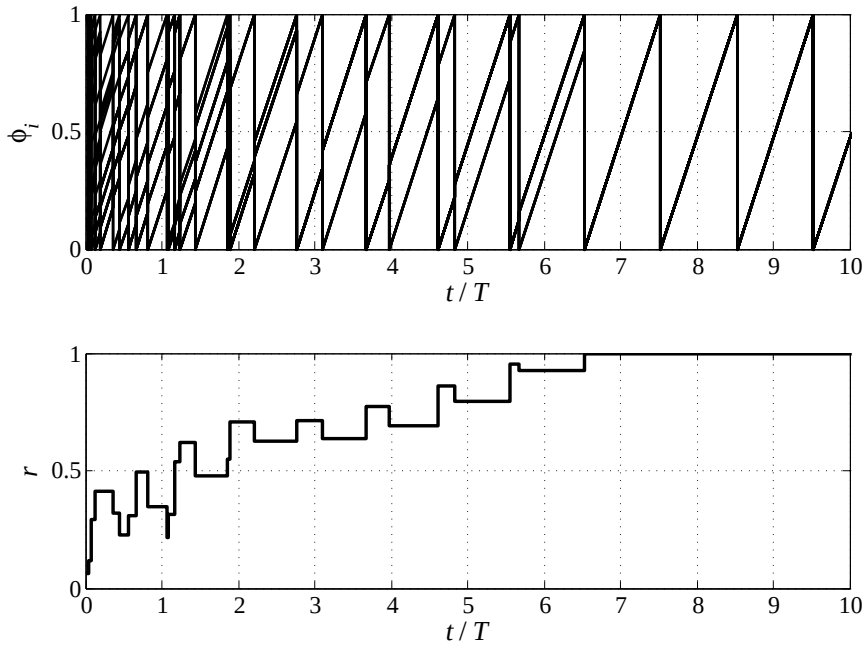


Figure 3.14: Evolution of phases ϕ_i over time and the corresponding Kuramoto synchronization index r.

From Figure 3.14 the synchronization index r starts with a value close to 0, and jumps abruptly when a node fires, this jump being proportional to the number of firing nodes. After three periods, two groups of oscillators have formed, at which point the metric is close to 0.6. Then the metric r gradually converges to 1, and synchronization is reached after six periods. This evolution would be more difficult to evaluate with the time representation of phases (top of Figure 3.14). Therefore the index captures the degree of coherence between oscillators in a simpler manner than the evolution of phases.

3.5.2 Local Synchronization

The Kuramoto index (3.22) is defined for the whole network. To grasp local interactions, a metric targeted at meshed network is defined, so that the synchronization state of a node with regards to its neighbors can be detected. This metric is applied to two case studies to illustrate its behavior in a meshed network compared to the Kuramoto synchronization index.

Local Metric

The state of a node is conveniently represented by its clock c_i in (3.2) in terms of the complex exponential of its phase. The proposed local synchronization metric for node i is defined as the

normalized sum of pairwise differences of local clocks:

$$r_i = 1 - \frac{1}{|\mathcal{N}_i|} \left| \sum_{j \in \mathcal{N}_i} (c_i - c_j) \right| . \tag{3.24}$$

The local metric is defined in the interval $r_i \in [-1, 1]$.

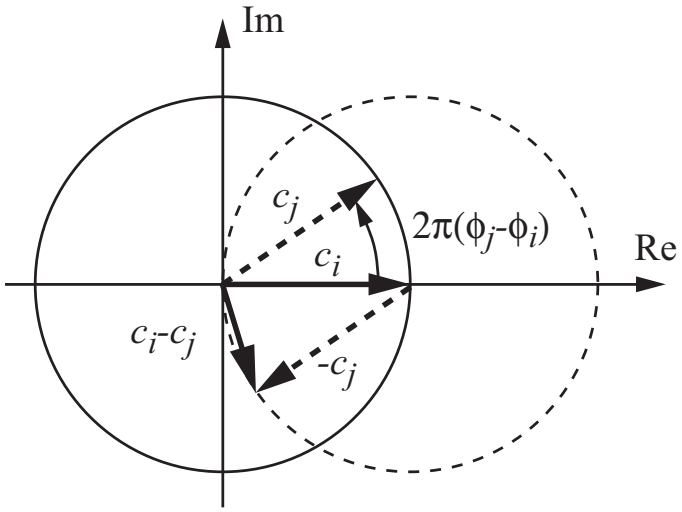


Figure 3.15: Geometric representation of two clock vectors c_i and c_j and their resulting difference $c_i - c_j$.

It is instructive to study the local metric by the evolution of the clock difference $c_i - c_j$ in the complex plane. Figure 3.15 shows the two clock vectors c_i and c_j and their resulting difference $c_i - c_j$, which constitutes one element of the sum in (3.24). In Figure 3.15 the phase of node i is equal to $\phi_i = 0$. The set of possible resulting clock differences $c_i - c_j$ describes a circle, which is plotted with a dashed line in Figure 3.15.

If node i is synchronized with all its neighbors, then $c_i = c_j$, $\forall j \in \mathcal{N}_i$ so that their difference is equal to 0, and the local metric yields $r_i = 1$.

Two interesting special cases result in $r_i = 0$. If neighboring nodes in $\mathcal{N}_i$ form two groups of equal size and are anti-phase synchronized, i.e. the firing instants of each group are delayed by $T/2$, then the pairwise difference in the sum over j with c_i in (3.24) gives 0 in half the sum and 2 in the other half, resulting in a total metric of $r_i = 0$. The metric also yields $r_i = 0$ when phases of neighboring nodes are distributed equally in $[0, 1]$, i.e. $\phi_j = j/|\mathcal{N}_i|$ for $j \in \mathcal{N}_i$. In this case, the pairwise differences with the clock of node i cancel each other, so that the normalized sum in (3.24) becomes 1, which results in $r_i = 0$. Therefore r_i is an appropriate metric for characterizing the local synchronization state.

The local metric is equal to $r_i = -1$ when two anti-phase synchronized groups form, one composed of node i only and all its neighbors forming the second group. Then $c_i = -c_j$, $\forall j \in \mathcal{N}_i$, so that the magnitude of the clock difference becomes $|c_i - c_j| = 2$, which yields $r_i = -1$.

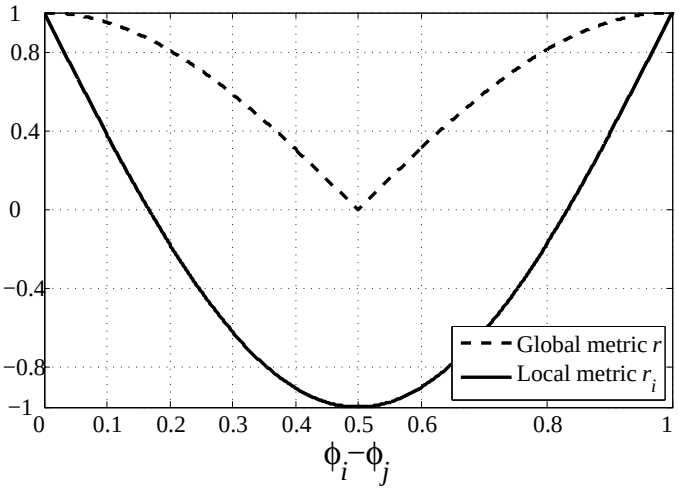


Figure 3.16: Resulting global and local metrics as the phase difference $\phi_j - \phi_i$ varies in a system of $N = 2$ nodes.

The resulting metric for $N = 2$ nodes is $r_i = 1 - |c_i - c_j|$, which yields, using the law of cosines:

$$r_i = 1 - \sqrt{2\big(1 - \cos(2\pi(\phi_j - \phi_i))\big)}\,. \tag{3.25}$$

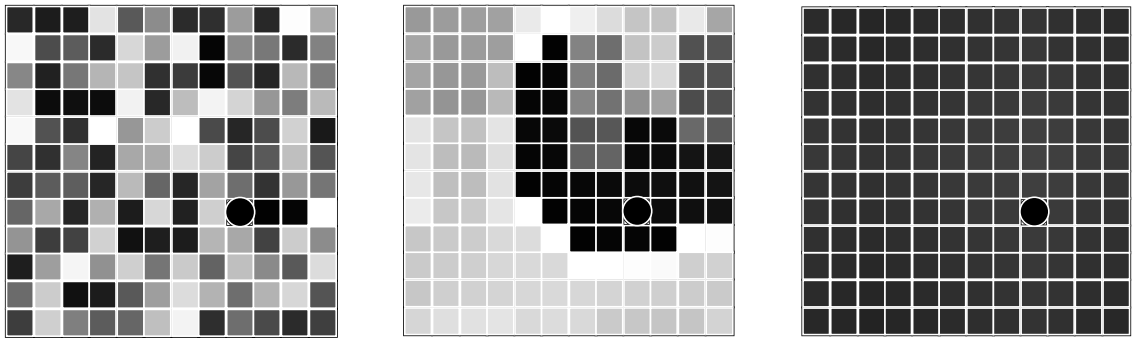
Figure 3.16 compares the Kuramoto global metric (3.23) and the proposed local metric (3.25) as the phase difference $\phi_j - \phi_i$ varies in a system of $N = 2$ nodes. From Figure 3.16 the Kuramoto metric is less penalizing against small phase differences. On the other hand, the proposed local metric decreases rapidly as the phase difference increases. For instance, when $\phi_j - \phi_i = 0.1$, the local metric yields $r_i \approx 0.4$, whereas the global metric yields $r_i \approx 0.95$. Around the maximum phase difference at $\phi_j - \phi_i = 0.5$, the slope of the global metric is discontinuous, whereas the local metric has a smooth trajectory. Therefore the local metric pronounces small phase changes but hardly varies when the phase difference is maximal.

A further difference between the proposed metric and the Kuramoto metric is that the former is exclusively computed based on local information, as the sum in (3.24) depends only on neighboring clocks. This is particularly useful when applying the metric to self-organizing networks where only local information is directly accessible. Therefore it is a powerful tool for exploring the local dynamics of pulse-coupled oscillators.

The proposed metric can be conveniently reformulated as a function of the degree and Laplacian matrices. Let $\mathbf{C}$ be defined as the vector of all internal clocks $\mathbf{C} = [c_1 \dots c_N]^{\mathrm{T}}$, where $[.]^{\mathrm{T}}$ is the transpose operator. Then the local synchronization metric can be written as:

$$\mathbf{R} = 1 - \Big|(\mathbf{D}(\mathcal{G}))^{-1} \cdot \mathbf{L}(\mathcal{G}) \cdot \mathbf{C}\Big| \tag{3.26}$$

with $\mathbf{R} = [r_1, \dots, r_N]^{\mathrm{T}}$.



(a) $t = 0T$ (b) $t = 5T$ (c) $t = 12T$

Figure 3.17: Representation of the phases of a grid of 12×12 oscillators at three instants during synchronization. The color of each node is proportional to its phase value, black corresponding to $\phi_i = 0$ and white corresponding to $\phi_i = 1$.

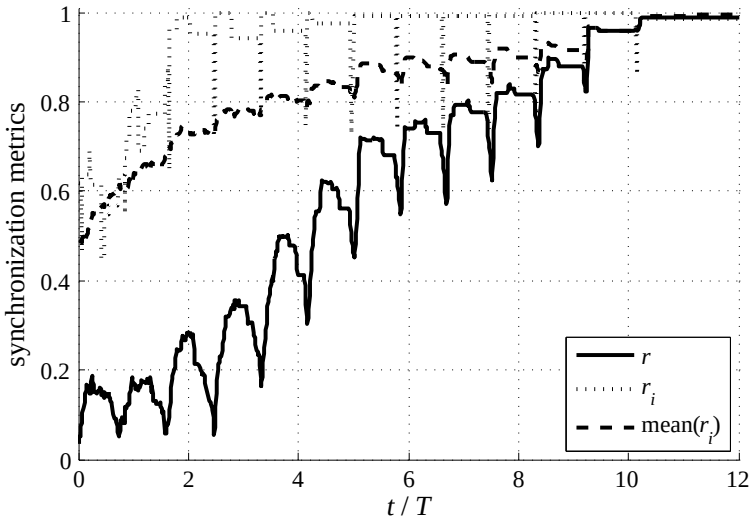


Figure 3.18: Evolution over time of the global and local synchronization metrics for a grid topology.

Case Study: Grid Network

The local metric is applied to a grid topology of 12×12 nodes shown in Figure 3.17. Nodes are coupled to their four closest neighbors, and wrap-around is considered so that all nodes have four neighbors. The coupling is delayed by $0.01\,T$ among neighboring nodes.

Figure 3.17 presents three snapshots of the network. The phase value of each node is displayed by a color, with black corresponding to $\phi_i = 0$ and white corresponding to $\phi_i = 1$. From the three snapshots, the network begins in a random state (Figure 3.17(a)), then pockets of synchronized nodes form after 5 periods in Figure 3.17(b), and eventually all nodes synchronize (Figure 3.17(c)).

The global and local synchronization metrics are applied to evaluate the evolution of the grid of oscillators. The local metric is considered for node i depicted as a circle marker in Figure 3.17.

Figure 3.18 plots the evolution of the Kuramoto metric r, the local metric r_i for node i , and the mean of all local metrics. In Figure 3.18 the local metric r_i rapidly converges to a value close to 1. This can be verified in Figure 3.17(b); node i is inside a cluster of nodes that have a similar color, and is thus synchronized with its four neighbors. On the other hand, the global Kuramoto metric does not converge to 1 before all nodes are synchronized, and its progression is much slower than the mean of local metrics. Due to the coupling delay among nodes, none of the metrics reaches exactly 1 when the network is synchronized. However as the delay is relatively small compared to the period duration, the metrics are close to 0.98.

Case Study: Ring of Oscillators

The *diameter* of the network is a common measure to describe a given topology (see Section 2.4); it is equal to the maximum number of links on the shortest path between any node pairs. To compare the behavior of the global Kuramoto metric and the proposed metric, N oscillators form a ring and communicate directly with the two nodes that have the smallest physical distance. The diameter of such a topology is equal to $N/2$ when N is even and to $(N-1)/2$ when N is odd. Figure 3.19 plots the evolution of the average global and local metrics over time for $1,000$ sets of initial conditions. The coupling parameters are set to $\alpha = 1.2$ and $\beta = 0.01$, and there is a constant interaction delay of $0.01\,T$ between two neighboring nodes.

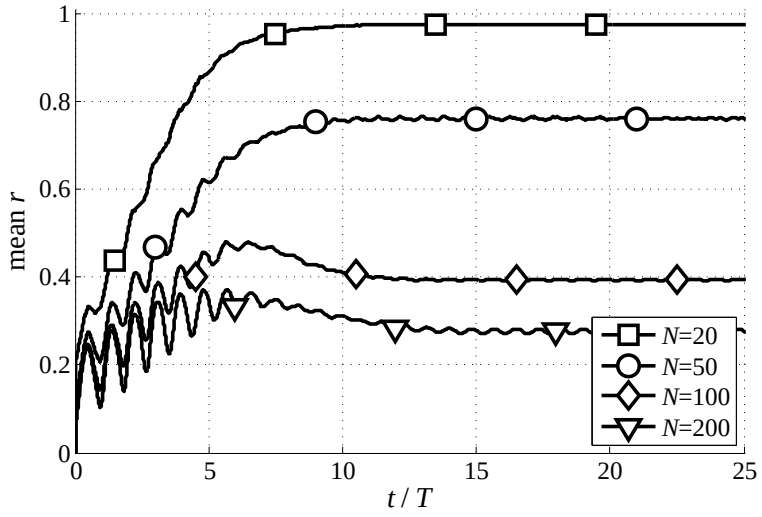
Increasing the network diameter impacts heavily the global metric. For $N = 20$, the global metric converges to a value close to 1, which indicates that the whole network has approximately the same phase value. As the diameter increases, the global metric oscillates before settling to a value decreasing inversely proportional to N. This oscillating behavior is due to the constant delay between nodes [BCdS97]. On the other hand, the local metric does not change as N varies, which confirms that nodes are synchronized on a local scale. This is a spectacular result demonstrating a key benefit of self-organized synchronization: nodes synchronizes rapidly with their neighbors and the size of the network has little impact on the local behavior.

3.6 Behavior of the PCO Model Under Imperfect Dynamics

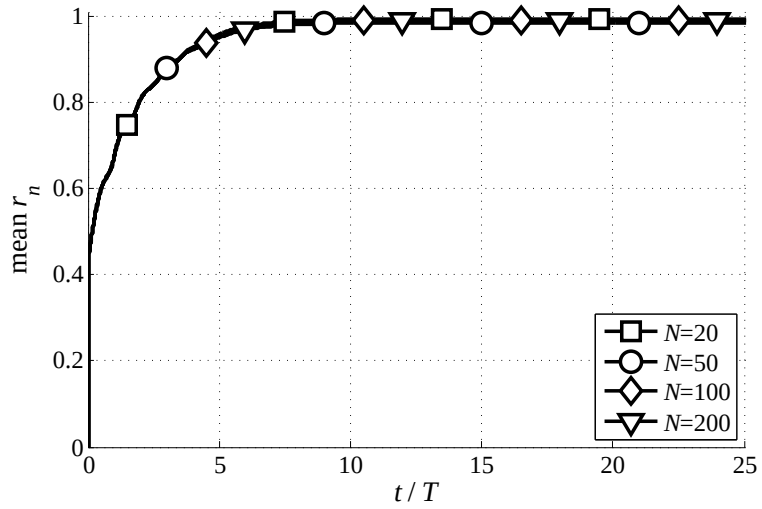
The pulse-coupled oscillator model presented in Section 3.1 assumes perfect conditions: oscillators have exactly the same dynamics and interact instantly with all others. In this section, two imperfections that are relevant for the application of the PCO model to wireless system are considered.

Discrepancies in the frequency of internal oscillators is unavoidable in practice (see Section 2.2.1), and in the following it is shown that even in meshed networks and within certain limits on the coupling, PCOs are still able to synchronize in the presence of frequency drift.

Section 3.4 showed that delays impacts the achieved accuracy. Propagation delays need to be taken into account when applying the PCO model to wireless networks, and have been shown to cause instability in the network [EPG95a]. This result is extended to meshed networks.



(a) Global metric



(b) Local metric

Figure 3.19: Evolution over time of the mean global and local metrics in a ring of N oscillators.

Both imperfections are studied assuming that oscillators form a meshed network. The synchronization metric is evaluated numerically in both cases, and enables to characterize the emergent state of PCO synchronization.

3.6.1 Frequency drift

A common question in network synchronization is to determine the robustness of a scheme with regards to frequency drifts. In (3.1) the phase of all nodes evolves with constant rate $1/T$, but in practice, two oscillators are never identical, and their internal frequency is randomly distributed around some nominal value. In this section the effect of frequency drift on meshed networks of

pulse-coupled oscillators is examined. This study is facilitated by the introduction of the local synchronization metric presented in Section 3.5.

In the presence of frequency drift, the internal clock rate of node i (3.1) becomes:

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}t} = \frac{1+\zeta_i}{T} \tag{3.27}$$

where ζ_i, the frequency drift of node i, is uniformly randomly distributed in $[-\zeta_{\max}, \zeta_{\max}]$ with $\zeta_{\max} > 0$. Thus natural periods of uncoupled oscillators are distributed between $T_{\min} = T/(1+\zeta_{\max})$ and $T_{\max} = T/(1-\zeta_{\max})$.

To compensate for the discrepancy of natural periods, nodes rely on the coupling with neighboring nodes. From the phase response curve given in (3.3), a phase increment is always strictly positive. Therefore nodes always shorten their period and advance their next firing instant, trying to catch up with the received pulse. In the presence of frequency drifts nodes thus try to catch up with the quickest oscillator, i.e. the one with shortest natural period.

If the frequency drift is too severe, some nodes may not able to catch up with the quickest oscillator. All receiving nodes are absorbed by the pulse of the fastest oscillator if their phase is in the interval $[\phi_\ell, 1]$. This puts a condition on the slowest oscillator, which should have a phase $\phi_{\min}$ which is at least equal to ϕ_ℓ when the quickest oscillator fires, as shown in Figure 3.20. From Figure 3.20, the minimum phase is equal to $\phi_{\min} = T_{\min}/T_{\max} = (1-\zeta_{\max})/(1+\zeta_{\max})$, which yields the following condition on the maximum clock drift:

$$\zeta_{\max} \leq \frac{1-\phi_\ell}{1+\phi_\ell}\,. \tag{3.28}$$

The absorption of slower nodes by the quickest one repeats itself periodically, and re-aligns all phases every period. Therefore, once nodes have synchronized, the synchronization metric stays close to 1 as long as condition (3.28) is met. To verify that this condition is valid in meshed networks, Figure 3.21 plots the evolution over time of the mean local synchronization metric for several values of $\zeta_{\max}$ obtained in networks of $N = 25$ nodes with a normalized algebraic connectivity

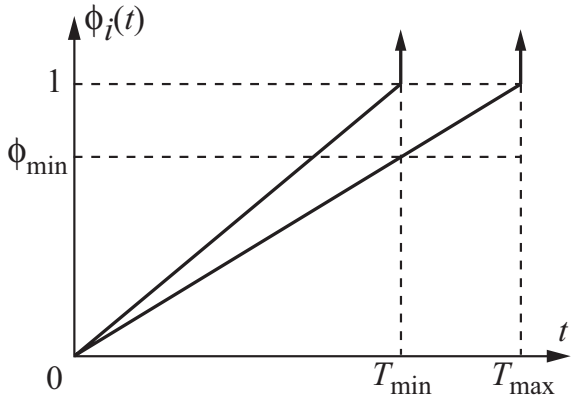


Figure 3.20: Minimum phase requirement on the slowest oscillator so that it is absorbed by the quickest oscillator at each firing instant.

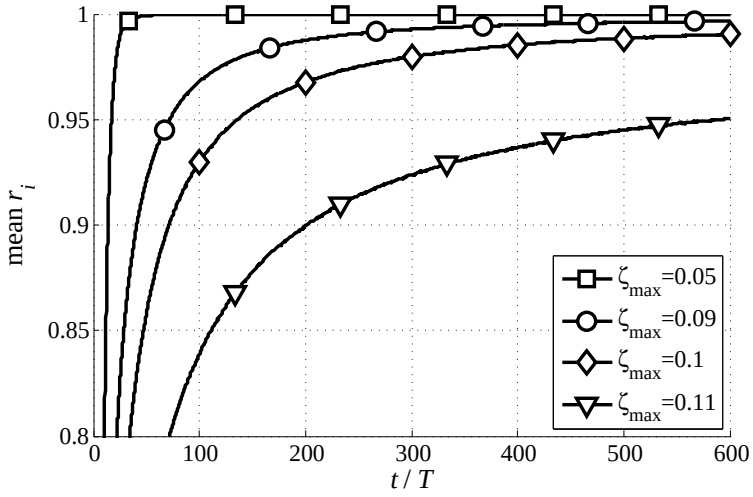


Figure 3.21: Time evolution of the mean local synchronization metric in the presence of frequency drift.

of $\kappa/N = 10^{-2}$.

Under the condition $\alpha = 1.2$ and $\beta = 0.01$, the maximum clock drift from (3.28) is equal to $\zeta_{max} \approx 0.10$. In Figure 3.21 as long as the drift is below this threshold, nodes are able to synchronize, but the time to synchrony augments as the drift increases. When the drift threshold is exceeded, i.e. $\zeta_{max} = 0.11$, a large gap in the synchronization metric is observed. Nevertheless the system is more ordered than the initial condition, because a few nodes only are not able to follow the quickest one.

3.6.2 Delays

When applying the PCO model to wireless networks, propagation delays delay the coupling between nodes. It was shown in [EPG95a] that coupling delays may lead to instability. By the introduction of a refractory period where phase adjustments are not permitted, stability is regained. In this section unstable synchronization states and the effect of a refractory period are characterized by the local synchronization metric.

Determining the *state* of a system of PCOs in the presence of coupling delays is a difficult task. In a fully-meshed network where all nodes are coupled directly to all others, the two node case discussed in Section 3.4.1 can be generalized as follows: if the refractory duration is at least equal to twice the maximum propagation delay, the first firing node forces its delayed firing instant onto other nodes, because it discards their echoes. Echoes are discarded if ϕ_{refr}, the refractory duration common to all nodes, satisfies the condition:

$$\phi_{refr} > 2 \max_{(i,j)\in\mathcal{E}} \frac{\nu_{ij}}{T} \tag{3.29}$$

where ν_{ij} accounts for the propagation delay between node i and its neighbors j.

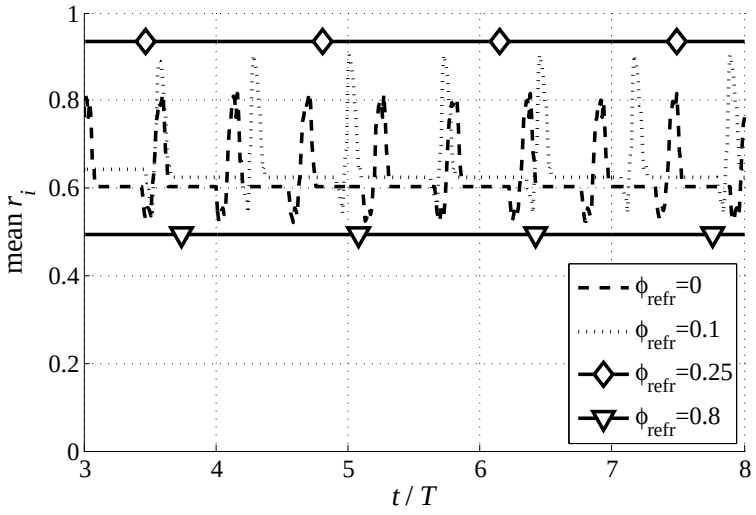


Figure 3.22: Example of the time evolution of the synchronization index for different refractory durations.

In a meshed network, the achieved stable state is not as clear, because the first firing node does not directly influence all other nodes. To the best of our knowledge, the impact of propagation delays in this case is an unsolved problem. The local synchronization metric is thus applied to examine whether the stability condition (3.29) is valid. As an example, Figure 3.22 plots the evolution of the metric for several values of the refractory duration in a network of $N = 12$ nodes with connectivity $\kappa/N = 0.5$ and a maximum propagation delay of $\nu_{max} = 0.1\,T$.

The synchronization metric in Figure 3.22 displays different behaviors depending on the duration of the refractory period. If ϕ_{refr} is too short and (3.29) is not met locally, then r_i oscillates, which corresponds to an unstable state as phases never align. If the refractory period is too large, i.e. $\phi_{refr} > 0.5$, nodes are not able to synchronize, and the metric remains low. Finally, if the duration of refractory is appropriately chosen, e.g. $\phi_{refr} = 0.25$, nodes synchronize, and the metric displays a constant value that is close to $r_i = 1$. The difference $1 - r_i$ accounts for the effect of the achieved accuracy, which is bounded by the propagation delays.

To generalize these results, Figure 3.23 plots the achieved mean synchronization index in the stable state for different maximum propagation delays as the refractory duration ϕ_{refr} augments, where ϕ_{refr} is common to all nodes. Simulations are conducted in networks of $N = 25$ nodes with $\kappa/N = 10^{-2}$.

Figure 3.23 confirms that the stability condition (3.29) is valid in meshed networks. In case $\phi_{refr} < 2\nu_{max}/T$, the refractory duration is too short, so that the system is unstable, expressed by a low synchronization metric r_i. For $2\nu_{max}/T \leq \phi_{refr} < 0.5$, the synchronization index is maximum and remains constant. Therefore the achieved accuracy does not depend on the refractory duration ϕ_{refr} but only on propagation delays. Finally, when $\phi_{refr} > 0.5$, the synchronization index drops due to deafness between nodes, i.e. nodes spend more time in refractory than in listen, and when nodes are only in refractory ($\phi_{refr} = 1$), the synchronization metric is minimal, as nodes remain in

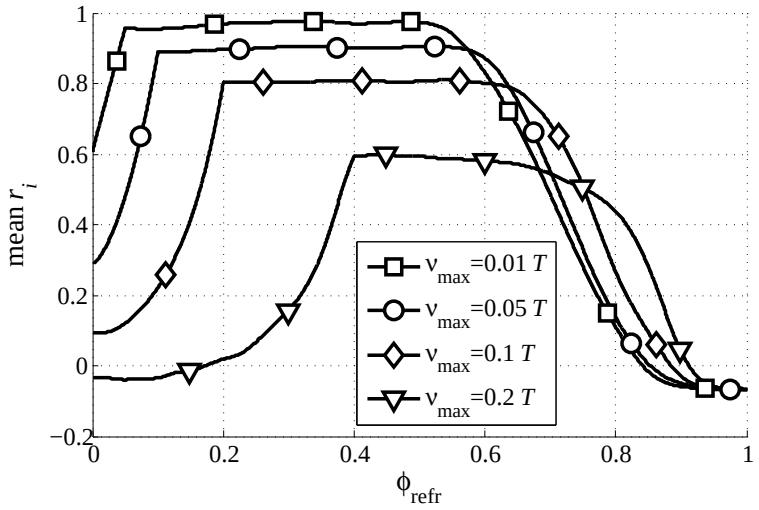


Figure 3.23: Mean synchronization index for different values of the maximum propagation delay.

the initial unordered state.

3.7 Application of PCO Synchronization to Wireless Networks

The PCO model provides simple rules leading to synchronization, and has been applied to different fields, integrating different constraints.

A straightforward application of PCO dynamics is the study of neural networks. Each neuron is modeled as an oscillator that emits electrical impulses periodically, and adjusts its emission instant when receiving impulses from other neurons. The phase model for PCOs enabled Izhikevitch [Izh99] to study neural systems and to show possible behaviors, e.g. synchronization if neurons have roughly the same internal frequency, or oscillatory associative memory behavior under certain coupling assumptions. The popularity of this model led to a special issue on Pulse Coupled Neural Networks in the IEEE Transactions on Neural Networks in May 1999 (Volume 10, Issue 3).

PCO dynamics have also been applied in image processing to design a computationally efficient segmenting algorithm. In [RF01] the PCO model is applied to perform image clustering. A coupling function different to (3.4) is defined that organizes a population of PCOs into clusters: within each cluster oscillators fire synchronously, while clusters themselves fire with a constant phase difference.

The application of PCO synchronization and dynamics to wireless systems is particularly attractive, especially for ad hoc and wireless sensor networks, as it enables networks to synchronize and self-organize in a distributed manner. It further feels intuitive to apply the PCO scheme to wireless networks given the broadcast nature of the medium. Various implementations and adaptations to wireless networks have been considered and are described below.

3.7.1 Direct Adaptations

The synchronization proof of Mirollo and Strogatz spawned a variety of studies on the PCO model. Some studies extended the PCO rules considering nodes communicate through discrete pulses.

Among the first to consider its application to wireless networks, Mathar and Mattfeld studied the impact of propagation delays [MM96]. A formal proof for two nodes in this case was presented and derived the results summarized in Section 3.4.1. The effect of a refractory period on the dynamics was further studied by Kirk and Stone in [KS97], who derived the firing map in the presence of a refractory period. Another important extension of the PCO model was presented by Lucarelli and Wang in [LW04]; it showed, under weak coupling assumption, i.e. $\alpha - 1 \ll 1$ and $\beta \ll 1$, that nodes asymptotically converge when they form a meshed network. This proof used results from consensus theory and from the derivation of phase deviation variables. This approach is further elaborated in Section 4.3.

3.7.2 Adaptations Based on Timing Detection

Practical applications of the PCO model placed on the physical layer of transceivers have been proposed. Placing the synchronization unit on the physical layer enables to minimize coupling delays, which in turn minimizes the achieved accuracy.

Hong and Scaglione in [HS03, HS05] utilized the characteristic pulse of Ultra Wide Band (UWB) radio to emulate the PCO synchronization principle. Bush [Bus05] examined the energy efficiency of the PCO model in a wireless environment, and showed greater efficiency and robustness with regards to mobility over classical time synchronization techniques. Recently Simeone *et al.* [SSBNS08] built upon the broadcast nature of wireless systems to superimpose transmitted shaped pulses.

3.7.3 Adaptations Based on the Exchange of Packets

Finally some adaptations of the PCO model rely on the exchange of data packets containing timing information in the form of timestamps in order to perform synchronization.

Werner-Allen *et al.* proposed the Reachback Firefly Algorithm: the synchronization unit is placed on the MAC layer, and synchronization is performed through the exchange of a low-level timestamp [WATP$^+$05]. Daliot [DDP03] proposed a self-stabilizing fault-tolerant algorithm named the Pulse Synchronization algorithm. The algorithm attains near optimal synchronization tightness while tolerating up to a third of the nodes exhibiting Byzantine behavior concurrently.

The PCO model has also been considered to achieve goals different from synchronization, because it provides a solid framework for designing self-organized algorithms. Wakamiya *et al.* applied the PCO model for data gathering and a traveling wave based communication mechanism for wireless sensor networks [KWM04, TWM07]. The implementation on wireless motes was made, and the algorithm coped with message delay and missed packets by selecting a subset of neighboring nodes. Degesys *et al.* modified the PCO rules to design an algorithm that achieves a round-robin

schedule in a decentralized manner [PDN07]; nodes fire one after the other with a constant offset. Babaoglu [BBJ07] applied the PCO model to overlay networks. These networks build upon IP networks by abstracting them.

3.8 Summary

This chapter presented the simple local rules forming the PCO model and studied its convergence for two nodes. Convergence to a synchronized system was shown by studying the return map, which iteratively always bring the phases closer, except for an unstable fixed point. For many nodes the proof of Mirollo and Strogatz relies on a similar approach. The dimension of the return map is reduced each time an absorption occurs, i.e. when the firing node forces some receiving nodes to join its firing instant.

The PCO model was extended to relieve some constraints made in the original demonstration and to measure its robustness to imperfections. The characteristics of the PCO model are conserved in the adaptations presented in subsequent chapters, and the present chapter presented a clear picture of its advantages regarding robustness, further to its simplicity.

When oscillators form a meshed network and are directly coupled to a subset of oscillators, the PCO model still leads to synchronization, but the coupling should remain sufficiently low and becomes an important parameter. This can be explained by the presence of nodes that bridge two parts of the meshed network: for very high coupling receiving nodes instantly adopt the timing of a received pulse, i.e. absorptions are immediate, which implies that a node bridging two unsynchronized parts of the network constantly updates its timing based on either side of the network. Convergence is obtained faster by letting nodes progressively update their clock, rather than immediately adopting the timing of a detected pulse.

Accounting for delays into the PCO model requires to introduce a refractory period after firing: pulses received in this period do not update the phase function. This enables to regain a stable system, but nodes no longer fire perfectly in-phase. The timing difference in firing instants defines the achieved accuracy, and it was shown through simulations that this accuracy among neighboring oscillators is never larger than the direct coupling delay. The accuracy among non-neighboring oscillators is influenced by the network topology, and improves for higher connectivity values, e.g. for normalized connectivities equal or higher than $\kappa/N \geq 10^{-2}$, the accuracy is equal or lower than the coupling delay in 90% of cases.

Studying the dynamics of a system of PCOs was facilitated by introducing a synchronization metric. This metric bridges the separate studies on the emergence of synchrony, i.e. the time to synchrony, and the steady state of the system, i.e. the achieved accuracy, and forms a comprehensive picture of the evolution of the system by quantifying the local synchronization state. This metric was applied to study the impact of clock drift, which is a typical imperfection. It showed that even in sparsely connected networks and local drifts distributed 10% around the nominal frequency, synchronization emerges from any random misalignment. A second study revisited the introduction

of delays in the PCO model to study the dynamics and the introduction of a refractory period. The stability condition on the necessary refractory duration that was derived in [MM96] for fully-meshed networks was verified in meshed networks thanks to the local metric.

CHAPTER

4 Emergent Slot Synchronization in Wireless Networks

The PCO model that was presented and studied in Chapter 3 is a powerful tool for describing and analyzing the dynamics of natural phenomena such as firefly synchronization [SS93], communication among neurons [BC00], and earthquakes [HH95]. This chapter presents and evaluates a novel network slot synchronization scheme, termed Meshed Emergent Firefly Synchronization (MEMFIS[1]), which is based on the PCO model. MEMFIS enables to synchronize nodes in wireless networks in a self-organized manner, whilst retaining the advantages of the PCO model such as the ability to cope with any random initial misalignment and an achieved accuracy among neighbors equal or lower than the propagation delay.

MEMFIS features a number of differences compared to existing adaptations of the PCO model. The existing approaches summarized in Section 3.7 make a clear distinction between an *acquisition phase* where the network synchronizes to a common slot structure, and a *communication phase* where nodes transmit and receive data. However, one key requirement for self-organized networks is to flexibly adapt to an ever changing environment. To this end, changes in the network topology, where previously disconnected parts of a network merge, force the network to re-synchronize, i.e. the acquisition phase needs to be repeated on a regular basis. The proposed synchronization scheme relies on the detection of a common synchronization sequence that is transmitted along with data. Therefore no interruption in data exchange is needed.

Other synchronization schemes that are applicable for slot synchronization are approaches that utilize sophisticated hierarchical protocols [GKS03, DBR06]. In self-organized networks a hierarchy is to be established prior to synchronization. This involves exchange of messages and therefore imposes additional overhead. In contrast, for MEMFIS the only overhead is the time-multiplexed synchronization word (sync-word), which is always needed for reliable detection of messages.

A further key advantage of MEMFIS is that no explicit timing information in the form of timestamps is required. This reduces the signaling overhead, and avoids sophisticated protocols. To this end, the network synchronization unit can be implemented within the physical layer, minimizing the uncertainty about the transmitter access time and the receiver processing delay, which severely

[1] Results in this chapter have been presented in parts in [TAB06b, TAB09].

affect the accuracy of protocols relying on timestamps [SY04]. Furthermore, as all nodes use a common synchronization word, its transmission is not subject to interference or collisions. In fact, all nodes transmitting synchronously a common word can help a faraway receiver synchronize with the rest of the network, because it receives the sum of all transmitted signals (known as the reachback problem) [HS06].

Before presenting the MEMFIS algorithm Section 4.1 points out the difficulties of directly applying the PCO model to wireless networks. Despite the simplicity of this model, a number of ideal assumptions are made in the proof of Mirollo and Strogatz [MS90] that cannot be neglected in wireless systems.

Integrating the constraints of wireless networks into the PCO model led to the slot synchronization algorithm MEMFIS, which is detailed in Section 4.2. The rules remain relatively simple, and the fundamental properties of the PCO model retained.

Before evaluating the performance of MEMFIS, two important points are addressed. In a similar manner to the analysis of the PCO model in the previous chapter, Section 4.3 analyzes the dynamics and the convergence of MEMFIS first for two nodes, before extending to many nodes. Secondly Section 4.4[2] examines the realistic detection of the common synchronization word employed in MEMFIS. Two types of receivers are considered, one based on cross-correlation of the received signal with the known synchronization word, and one based on auto-correlating the received signal and searching for a known pattern.

In Section 4.5 the performance of MEMFIS is evaluated against three key requirements, namely the scalability, the adaptability to the network connectivity, and the robustness in case of unreliable detection of synchronization words. Furthermore MEMFIS is compared to an alternative decentralized slot synchronization algorithm where clocks are adjusted based on the weighted average of received synchronization signals [AAK91]. Simulation results demonstrate that synchronization emerges faster, in particular as the network gets sparser.

MEMFIS does not depend on one standard and can accommodate any given slot structure. Section 4.6 presents two adaptations of MEMFIS to existing frame structures, namely adaptations to the IEEE 802.11 standard and to the Sensor-MAC protocol are presented.

4.1 Constraints of Wireless Networks

The PCO model described by Mirollo and Strogatz in [MS90] achieves synchronization from any initial condition assuming several ideal assumptions. When applying this model to a wireless network, some of these assumptions are not valid, and the model thus cannot be directly applied.

In Chapter 3, the dynamics of the PCO model were examined removing two important ideal assumptions from the original model of [MS90]:

- *Coupling delay:* Delay in the coupling causes instability and synchronization is not reached.

[2]The second part of this section is the result of a collaboration with Dr. Luca Sanguinetti from Pisa University, Italy, and have been published in parts in [STMA08]

This is tackled by introducing a refractory period after transmission, and enables to regain a stable system with an accuracy among neighboring nodes limited by the direct coupling delay.

- *Meshed network:* If nodes form a meshed network and are not coupled all-to-all, the PCO model still leads to synchronization. No formal proof is yet available showing synchronization for any coupling strength, but simulations show that synchronization is reached if the coupling is not too high (see Section 3.3).

Applying the PCO model to wireless networks requires to account for further constraints. To understand and illustrate the constraints of wireless networks on this model, Figure 4.1 examines a typical communication chain and presents the Open Systems Interconnection Reference Model (OSI Model) communication model between a transmitting node Tx and a receiving node Rx.

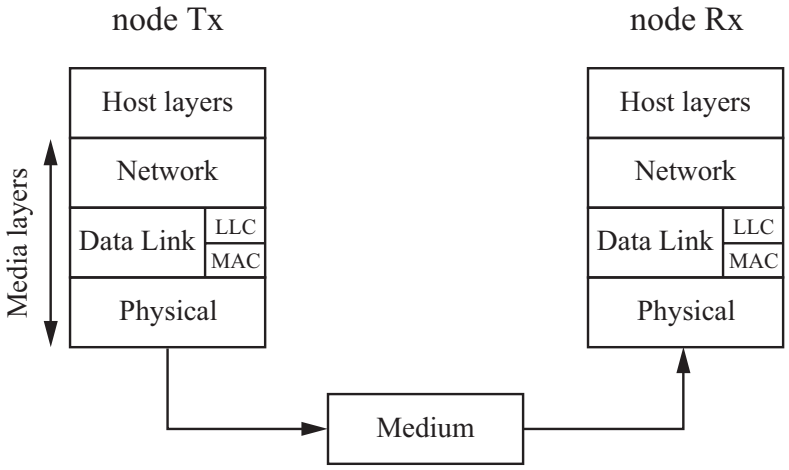


Figure 4.1: Decomposition in layers with the OSI Model for the communication between a transmitting node Tx and a receiving node Rx.

The OSI model is commonly used as an abstract description for layered communications and computer network protocol design. It divides the communication between two users into a stack of seven layers at each end, going from the physical transmission over a medium to the user level. The first three layers are the physical, the data link, and the network layers; they correspond to the media layers and are concerned with sharing a communication medium with other users and transporting the data from the transmitter's network layer to the receiver's network layer. The four remaining layers are the host layers, i.e. the transport, session, presentation, and application layers, and are in charge of delivering the information to the end user.

Communication between each layer in the OSI Model adds delays. As delays are critical for the stability and the achievable accuracy of the PCO Model, they should be minimized. Some delays are unavoidable, e.g. the propagation delay corresponding to the transmission through the medium. Therefore MEMFIS places the synchronization unit as closely as possible to the physical

transmission in order to minimize transmission delays. Further constraints from the physical layer and the MAC layer are examined below.

4.1.1 PHY Constraints

The Physical Layer (PHY) defines the physical specifications for transmission and reception. In particular, it defines the relationship between a device and a physical medium: it is in charge of transforming a received physical signal into a series of bits that are passed to the data link layer, and reversely, to generate and transmit a physical signal from the bits passed by the data link layer. The major functions performed by the PHY in digital wireless systems are:

- to send a physical signal between nodes Tx and Rx; this is accomplished by a number of signal processing operations such as up- and down-conversion, sampling, modulation, channel coding, channel estimation, symbol synchronization, to name a few;
- to participate in the resource sharing process such as contention resolution, e.g. carrier sensing, retransmission and combination of the received signals.

The PHY is thus in charge of establish an error-free communication channel between the two communicating nodes. The nature of a wireless channel and the processing required at the PHY imposes constraints on the PCO model that require two important modifications, namely the transmission of a long sync-word and a decoding delay at the receiver:

Transmission of Long Sequences In the PCO model, nodes interact through pulses. In a wireless environment, noise and interference are present and make the detection of pulses almost impossible. It is therefore more robust to consider a long synchronization word of duration T_{Tx}. As the synchronization process relies on identifying the timing and not the source, a synchronization word common to all nodes is considered.

Decoding Delay The signal processing operations at the PHY necessarily incur a delay from the instant a signal is received until the instant the common synchronization word is properly received and detected. This delay is referred to as decoding delay and is denoted T_{dec}.

4.1.2 MAC Constraints

The Data Link Layer provides the functional and procedural means to transfer data between nodes in the network, and to detect and possibly correct errors that may occur in the PHY. It is concerned with the interactions of multiple devices with a shared medium, whereas the PHY is concerned with the received physical signal.

The Data Link Layer is decomposed into two sublayers. The Logical Link Control (LLC) sublayer is in charge of error control through Automatic Repeat Request (ARQ) and of flow control. The Medium Access Control sublayer takes the bits from the PHY and forms a frame. It is in

charge of data packet queuing or scheduling, channel-access control, and physical addressing (MAC addressing).

As MEMFIS is placed on the physical layer, it needs to obey to certain constraints that are imposed by the MAC layer:

Slotted Transmission Given a time slotted protocol, time is divided in a given slot structure. Slots are decomposed into transmitting and receiving slots, each 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.

Time Slot Allocation The PHY dictates *how* to transmit, whereas the MAC decides *when* to transmit. 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.

4.2 MEMFIS: Meshed Emergent Firefly Synchronization

In this section an adaptation of the PCO model targeted at wireless networks is described. This adaptation is able to cope with realistic implementation assumptions and is designed to achieve slot synchronization in a self-organized manner.

4.2.1 Transceiver Architecture

Figure 4.2 depicts a block diagram of the MEMFIS transceiver architecture, consisting of a reception, synchronization, and transmission unit. The transceiver is half-duplex, i.e. reception and transmission cannot occur simultaneously. A Boolean state variable $s_i \in \{0,1\}$ indicates whether packets from the MAC layer are scheduled for transmission or not. In case $s_i = 1$, the transceiver switches to transmission mode at the next reference instant τ_i. Otherwise, when $s_i = 0$ the incoming signal is directed to the reception unit. Switching from one state to the other is possible only when the node fires at τ_i; this is ensured by the Sample and Hold (S/H) block in Figure 4.2.

In transmit state, data coming from the MAC layer of node i, denoted by the sequence $\{a_i[k]\}$ with the discrete time index k, are to be transmitted. The transmission unit multiplexes this sequence with the common sync-word $\{b[k]\}$ to form the transmitted symbol sequence $\{x_i[k]\}$. After digital-to-analog conversion (block "MOD"), the signal $x_i(t)$ including the sync-word $b(t)$ is transmitted over the wireless channel.

In receive state, incoming signals $y_i(t)$ are directed to the reception unit. The sync-word detector scans for sync-words $b(t)$ sent by a neighboring node j, and determines its reference instant τ_j. The estimate of τ_j at node i, denoted by $\widehat{\tau}_{ij}$, allows the reception unit to locate the received symbol sequence $\{\widehat{x}_j[k]\}$ within the incoming signal stream. While $\widehat{\tau}_{ij}$ is passed to the slot synchronization unit, the payload data of node j is decoded (block "DEC") to yield an estimate of the data packet $\{\widehat{a}_j[k]\}$, which is provided to the MAC layer.

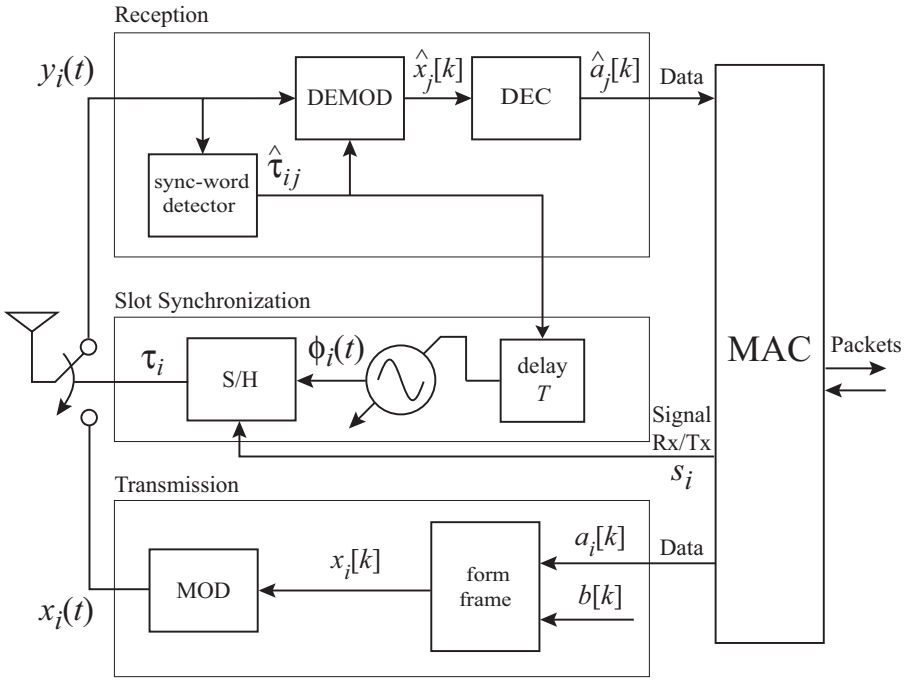


Figure 4.2: Block diagram of the MEMFIS network synchronization unit.

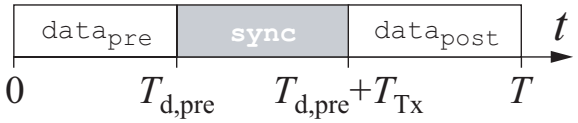
4.2.2 Transmit slot

As detailed in Section 4.1, a long synchronization word of duration T_{Tx} is considered as the synchronization mean. It enables to cope with noise and interference present in wireless networks, which is more robust than a single pulse. The common sync-word is embedded in each transmit slot, and enables receiving nodes to adjust their phase when detecting it. Similarly to the PCO model, the synchronization unit relies on identifying the timing and not the source of transmission, and therefore a synchronization word common to all nodes is used throughout the network.

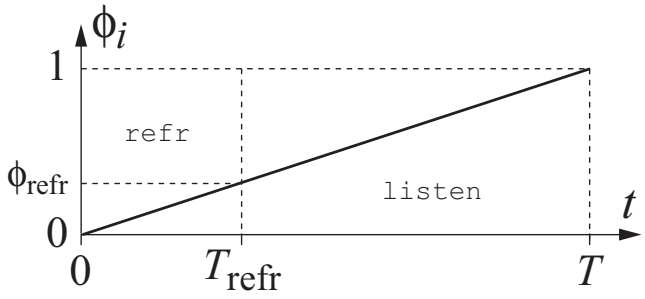
A considered transmit time slot is composed of the following parts: payload data are transmitted in states $\texttt{data}_{\texttt{pre}}$ and $\texttt{data}_{\texttt{post}}$ of respective durations $T_{\text{d,pre}}$ and $T_{\text{d,post}}$, and the synchronization word is transmitted in state `sync`. Durations of the three states are chosen such that:

$$T_{\text{d,pre}} + T_{\text{Tx}} + T_{\text{d,post}} = T \ . \tag{4.1}$$

Figure 4.3(a) illustrates a transmit time slot where the synchronization word is placed as a midamble, i.e. $T_{\text{d,pre}} > 0$ and $T_{\text{d,post}} > 0$. The structure of a transmit slot reflects a general packet structure of any slotted transmission and may be adjusted for a particular standard or MAC layer. For example, in the IEEE 802.11 standard [IEE99], each frame includes a preamble placed at the start of a slot. In this case $\texttt{data}_{\texttt{pre}}$ is omitted, so $T_{\text{d,pre}} = 0$. This case along with the S-MAC slot structure are detailed in Section 4.6.



(a) Transmit slot when the synchronization word is placed in the midamble



(b) Receive slot

Figure 4.3: Transmit and receive slots in MEMFIS.

4.2.3 Receive slot

When no packet needs to be transmitted, a node stays in receive slot to receive data and synchronization words from other nodes ($s_i = 0$). The structure of a receive slot is based on the PCO model described in Chapter 3, and is composed of a refractory period of length T_{refr} (no phase adjustment is possible) and a listening period of length T_{Rx}, as shown in Figure 4.3(b). In a receive slot a node maintains a phase function linearly increasing over time (3.1), but the phase is only adjusted during `listen`. Durations of the two states are chosen such that:

$$T_{\text{refr}} + T_{\text{Rx}} = T \ . \tag{4.2}$$

In the PCO model the refractory period is needed for stability, i.e. to avoid that echoes are acknowledged. In MEMFIS echoes are implicitly neglected thanks to the half-duplex assumption, i.e. a node firing and transmitting simultaneously with the transmitting node cannot be heard. Nevertheless a refractory period is useful in the listen slot because it reduces the time to convergence. This point is investigated in Section 4.5.5 through simulations; it is shown that the refractory period is particularly useful in networks with low connectivities, whereas its choice has little impact for higher algebraic connectivities.

During a receive slot, the sync-word detector scans for the known sync-word that is common to all nodes. The detection of this sequence is a very important part of the synchronization process; it should produce a series of peaks, each peak corresponding to the transmission of a neighboring node, in a similar manner to the pulses transmitted in the PCO model. These peaks are used to estimate the firing instants of transmitting nodes, and are passed on to the slot synchronization unit. Two detectors for the receive slot are further detailed and evaluated in in Section 4.4.

A transmission from node j produces a peak at the detector at instant θ_{ij}, which is equal to a

delayed version of τ_j, the firing instant of transmitting node j. This firing instant is delayed by:

- the propagation delay ν_{ij}: the time required for the signal transmitted by node j to reach the receiver of node i;
- the position of the sync-word $T_{\text{d,pre}}$: the transmission of the sync-word only starts at $\tau_j + T_{\text{d,pre}}$;
- the sync-word duration T_{Tx}: the detector needs to receive the complete sync-word before a synchronization peak occurs.

Therefore the detection peak instant θ_{ij} at receiving node i is equal to:

$$\theta_{ij} = \tau_j + \nu_{ij} + T_{\text{d,pre}} + T_{\text{Tx}} \ . \tag{4.3}$$

Without knowledge of the propagation delay ν_{ij}, the sync-word detector of node i substracts the known delays from the detection instant θ_{ij}, which results in the estimated firing instant of node j:

$$\begin{aligned} \widehat{\tau}_{ij} &= \theta_{ij} - T_{\text{d,pre}} - T_{\text{Tx}} \\ &= \tau_j + \nu_{ij} \end{aligned} \tag{4.4}$$

4.2.4 Slot Synchronization

The detected reference instants $\widehat{\tau}_{ij}$ are passed from the sync-word detector to the slot synchronization unit in Figure 4.2. While the introduction of long sync-words of duration T_{Tx} greatly improves the reliability to successfully detect sync-words and enables to transmit a sync-word along with data, it induces transmission delays. When applying the synchronization rules described in Chapter 3 without modifications, these delays drastically compromise the attainable slot synchronization accuracy [TAB06a].

A fundamental requirement to reach synchrony in the presence of transmission delays is to make the coupling delay among nodes equal to T [TAB06a]. Delaying the coupling by $T_{\text{d,post}}$, node i increments its phase at

$$\begin{aligned} \theta_{ij} + T_{\text{d,post}} &= \tau_j + \nu_{ij} + T \\ &= \widehat{\tau}_{ij} + T \ . \end{aligned} \tag{4.5}$$

Consequently, for node i in `listen` state (see Figure 4.3(a)), a phase increment is decided based on the correlator output (4.15) delayed by $T_{\text{d,post}}$:

$$\phi_i(t^+) = \begin{cases} \phi_i(t) + \Delta\phi(\phi_i(t)) \ , & s_i = 0,\ \phi_i(t) > \phi_{\text{refr}} \\ & \text{and } \Lambda_i(t - T_{\text{d,post}}) > \Lambda_i^{[\text{th}]}(t - T_{\text{d,post}}) \\ \phi_i(t) \ , & \text{elsewhere.} \end{cases} \tag{4.6}$$

where Λ_i is the correlator output of the sync-word detector and $\Lambda_i^{[\text{th}]}$ is the comparison threshold of the detector.

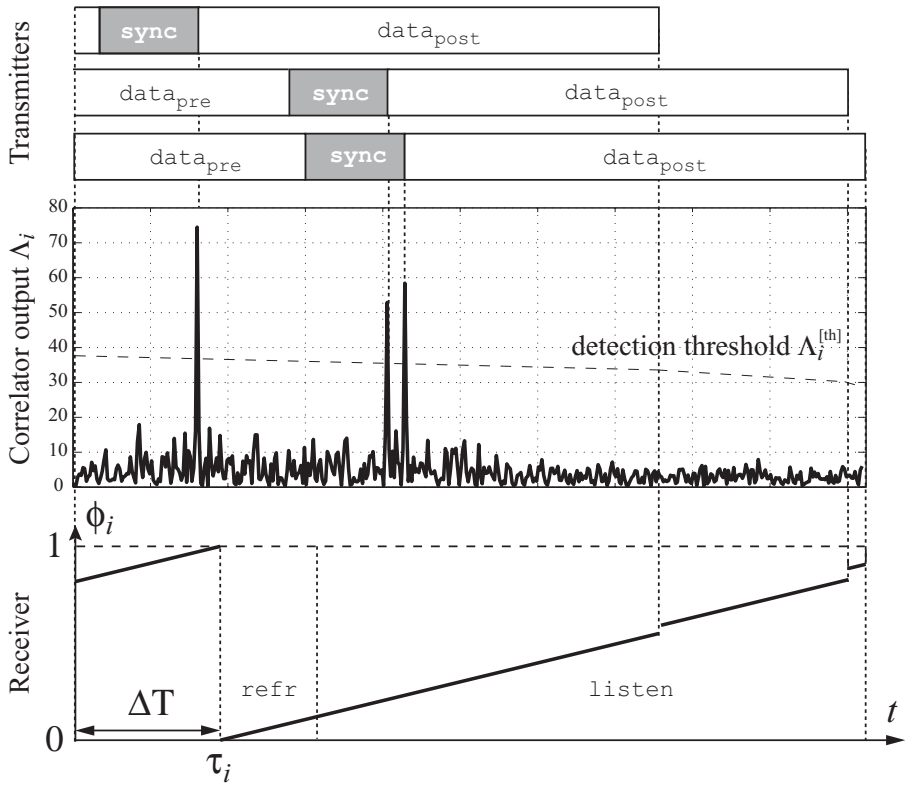


Figure 4.4: Example of a correlator output causing multiple delayed increments to the phase function of receiving node i.

According to (4.5) and (4.6), the delay in the coupling between transmitting and receiving nodes is equal to $T + \nu_{ij}$. This ensures that the slot synchronization accuracy is bounded only by propagation delays ν_{ij}, provided a sufficiently long refractory period (3.18). Notice that the induced delay $T_{\text{d,post}}$ in (4.6) also enables the receiver to compensate decoding delays with $T_{\text{dec}} \leq T_{\text{d,post}}$.

Figure 4.4 plots the output of the correlation detector of a receiving node i and its corresponding phase variable during one period. For simplicity propagation delays are neglected. During the represented period, three unsynchronized nodes transmit and cause three distinct increments of the phase variable. In Figure 4.4 receiving node i fires at $t = \tau_i$ and ends a listen period. As node i has no data to transmit another receive slot follows. The correlation detector produces a series of peaks at instances where transmission of sync-words is completed. As coupling of node i in (4.6) is delayed by $T_{\text{d,post}}$, the first peak that occurred at the correlator output during the previous time slot results in a phase increment $T_{\text{d,post}}$ later. As node i is in `listen` state at this instant, it increments its phase accordingly. Later the incidence of the two remaining sync-words is detected, which eventually causes node i to fire. After this process, node i has aligned its timing with the third transmitter that initially fired ΔT before i.

4.3 Convergence Analysis

Similarly to the synchronization analysis for the PCO Model in Section 3.2, the MEMFIS rules described in Section 4.2 are applied to a system of two nodes, and it is shown that despite the interaction delay of T, MEMFIS has the same firing map as the PCO Model, and thus similar dynamics. For an arbitrary number of nodes forming a meshed network, the synchronization proof of Lucarelli and Wang [LW04] is extended to account for interaction delays and random transmissions. In the remainder of this section, propagation delays are neglected for simplicity, and results are thus valid considering that the oscillation period T is much larger than the propagation delays ν_{ij}.

4.3.1 Two Nodes

To analyze the dynamics of MEMFIS, two nodes starting from an arbitrary condition are studied. This simple system is characterized by the phase functions ϕ_1 and ϕ_2 and the state variables s_1 and s_2. Without loss of generality, the system starts after node 1 has fired at $t = \tau_1$, and at this instant is in a state where both nodes are in a receive slot, i.e. $(\phi_1, \phi_2, s_1, s_2) = (0, \phi_0, 0, 0)$. This initial state is shown in Figure 4.5(a).

In Figure 4.5 both transmit and receive slots are represented on the same phase portrait. Nodes on the outer circle follow a receive slot ($s_i = 0$), and from the firing instant placed on right hand side of the portrait, go through a refractory period `refr` following by a listen period `listen`. Nodes on the inner circle follow a transmit slot ($s_i = 1$), composed of two data parts, $\texttt{data}_{\texttt{pre}}$ and $\texttt{data}_{\texttt{post}}$, and a sync-word `sync`. In Figure 4.5, node 1 is represented as a white marker, and node 2 as a black marker.

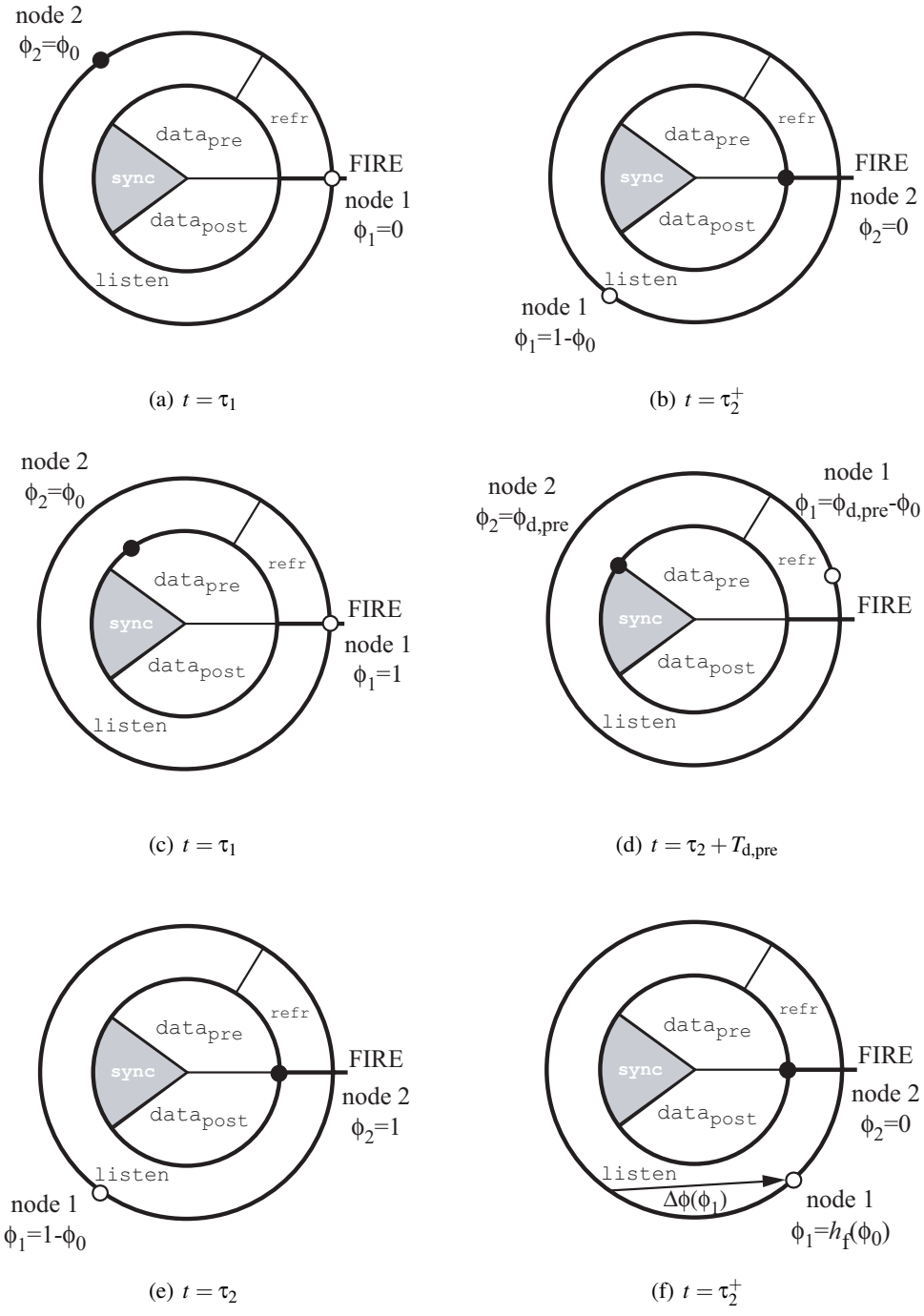
After the initial condition the following relevant event occurs when node 2 fires at $t = \tau_2$ (Figure 4.5(b)). At this instant node 2 enters a transmit slot, which results in a system state of $(\phi_1, \phi_2, s_1, s_2) = (1 - \phi_0, 0, 0, 1)$. As both nodes were in a receive slot before this event, the firing of node 2 has not influenced the phase of node 1.

Similarly the next event occurs when node 1 fires at $t = \tau_1$ (see Figure 4.5(c)), exactly T seconds after the initial condition depicted in Figure 4.5(a). The phase values have not changed between these two instants, but node 2 is now transmitting data in $\texttt{data}_{\texttt{pre}}$ state. After firing, node 1 remains in receive state, and enters another receive slot.

A short moment later, at instant $t = \tau_2 + T_{\text{d,pre}}$, node 2 starts transmitting the synchronization word (Figure 4.5(d)). Node 1 is in refractory at this moment, meaning that its receiver is switched on but its phase cannot change. The interaction between the nodes continues until node 2 fires at instant $t = \tau_2$ (Figure 4.5(e)). Node 1 has received the full sync-word from node 2, and after a further delay of $T_{\text{d,post}}$, it increases its phase from $1 - \phi_0$ by $\Delta\phi(1 - \phi_0)$. Following the definition of the firing map in (3.5), the firing map for MEMFIS is equal to:

$$\begin{aligned} h_{\text{f}}(\phi_0) &= \phi_1 + \Delta\phi(\phi_1) \\ &= -\alpha\phi_0 + (\alpha + \beta) \ , \end{aligned} \tag{4.7}$$

if $\phi_0 > 1 - \phi_\ell$, which is the same firing map as the PCO Model.



(a) $t = \tau_1$ (b) $t = \tau_2^+$

(c) $t = \tau_1$ (d) $t = \tau_2 + T_{\text{d,pre}}$

(e) $t = \tau_2$ (f) $t = \tau_2^+$

Figure 4.5: Synchronization of two nodes following MEMFIS.

Following a similar reasoning where nodes switch roles, i.e. node 1 transmits and node 2 receives, it is straightforward to show that the return map for MEMFIS is equal to the return map for the PCO Model (3.7). Therefore, although interactions with MEMFIS are delayed by one period,

the resulting system is the same as for the PCO Model. This is an important result for the extension to many nodes and the phase deviation transformation that follows, and it also implies that the results that were presented for the PCO Model on the achieved accuracy and the convergence in meshed networks under imperfect conditions (Sections 3.4 and 3.6) are also valid for MEMFIS.

4.3.2 Many Nodes

To formally prove that pulse-coupled oscillators synchronize from any initial condition is a non-trivial task. Formal derivations require a number of idealized assumptions. For example, Mirollo and Strogatz [MS90] ignore delays and assume that nodes form a fully-meshed network, i.e. all-to-all coupling between nodes. More recently the convergence to synchrony of the PCO model [MS90] was extended to meshed networks by Lucarelli and Wang [LW04], assuming weak coupling between oscillators. For their proof, the theory of phase deviation variables is utilized, which was introduced in [Izh99]. In the following the convergence of MEMFIS is studied based on phase deviation variables. Considering a meshed network, interactions between nodes following MEMFIS are conditioned on a given MAC policy and are delayed by exactly one period.

Phase Deviation

In the PCO model, oscillators maintain a phase, which grows linearly over time, and which instantly increments upon reception of pulses from neighboring oscillators. The evolution of each phase is thus equal to the sum of the uncoupled dynamics and the discrete interactions with its neighbors:

$$\frac{\mathrm{d}\phi_i(t)}{\mathrm{d}t} = \frac{1}{T} + \sum_{j \in \mathcal{N}_i} \left(\phi_i(t) + \Delta\phi\left(\phi_i(t)\right)\right) \cdot \delta(t - \tau_j) \tag{4.8}$$

where $\delta(.)$ is the Dirac function.

As all nodes have identical uncoupled dynamics, it is sufficient to study the evolution of the discrete interactions over time in order to prove synchrony. Averaging theory provides a change of variables that transforms the sum of *discrete* interactions between nodes in (4.8) into a sum of continuous interactions [Izh99]. The transformed variable is termed *phase deviation* φ_i. For the PCO model, phase deviations evolve according to [Izh99]:

$$\frac{\mathrm{d}\varphi_i(t)}{\mathrm{d}t} = -\frac{\varepsilon}{T\,2\pi} \sum_{j \in \mathcal{N}_i} \left(\varphi_i(t) - \varphi_j(t)\right) + O\left(\varepsilon^2\right) \tag{4.9}$$

where $\varphi_i \in [-\pi, \pi]$, $\varepsilon = \alpha - 1$, and $O\left(\varepsilon^2\right)$ is an asymptotic upper bound which decreases rapidly as a function of ε^2. This function can be neglected by assuming weak coupling, i.e. $\varepsilon \ll 1$.

The set of equation in (4.9) is more approachable than (4.8): each node updates its phase deviation as the sum of the difference between its own phase deviation and the one of neighboring nodes. This is a fundamental problem in consensus theory, and recent work in this field studied conditions for convergence [OSM04, Mor05].

By combining elements from graph theory, consensus theory and nonlinear dynamics, Lucarelli and Wang [LW04] showed for a general oscillator model where coupling is governed by a function ϑ and where phase deviations evolve according to $\mathrm{d}\varphi_i(t)/\mathrm{d}t = \varepsilon \sum \vartheta(\varphi_j(t) - \varphi_i(t))$, that oscillators always synchronize, i.e. $\varphi_i \to \varphi, \forall i$. The condition for convergence is that the coupling function ϑ is uneven, i.e. $\vartheta(0) = 0$ and $\varphi \cdot \vartheta(\varphi) > 0, \forall \varphi \neq 0$. From (4.9) the coupling function of the PCO model is equal to $\vartheta(\varphi) = \frac{1}{T \cdot 2\pi} \cdot \varphi$. Therefore, under the weak coupling condition, the coupling function of the PCO model fulfills the convergence criterion derived in [LW04], which proves its convergence in meshed networks.

Convergence of MEMFIS

MEMFIS features a number of differences to the PCO model, namely one period delay in coupling, no receiving whilst transmitting, and transmissions determined by the MAC. In the following discussion, perfect detection of sync-words is assumed, i.e. no false detection occurs.

Random Transmissions and Deafness In MEMFIS, synchronization messages are transmitted according to a given medium access scheme. Therefore interactions do not occur periodically as in the PCO model, but depend on the states of nodes. More precisely, two nodes can communicate not only if they are sufficiently close, but also if one node is receiving whilst the other transmitting. Therefore the edge set $\mathcal{E}(t)$ defining the network topology changes over time depending on the evolution of slot assignments from the MAC, and the graph representing the network is dynamic, i.e. it is defined by $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$. This impacts phase deviation variables (4.9) as follows:

$$\frac{\mathrm{d}\varphi_i(t)}{\mathrm{d}t} = -\frac{\varepsilon}{T\,2\pi} \sum_{j \in \mathcal{N}_i(t)} (\varphi_i(t) - \varphi_j(t)) \tag{4.10}$$

where $\mathcal{N}_i(t) \subseteq \mathcal{N}_i$ is the set of detected neighbors of node i at instant t. When node i is transmitting, no sync-word is acknowledged, and the set $\mathcal{N}_i(t)$ is empty, referred to as *deafness* between nodes. When node i is listening, $\mathcal{N}_i(t)$ is composed of its neighbors $\mathcal{N}_i$ that are in the set of transmitters $\mathcal{I}_i(t)$ and whose sync-word transmission is successfully detected.

Moreau [Mor05] showed that nodes following (4.10) converge if there exists a $T_c > 0$, such that within the time interval $[t, t + T_c]$ the union of topologies $\mathcal{G}(t)$ aggregates to a connected network, i.e. all nodes are mutually connected. Although this convergence condition is surprisingly weak, it does not imply that MEMFIS necessarily synchronizes from any initial condition. The MAC protocol directly influences the dynamic topology $\mathcal{G}(t)$ through the set of observable neighbors $\mathcal{N}_i(t)$. In fact, if the MAC protocol fails to ensure that the union of topologies is connected over time, synchronization may not always be reached. This is observed in the simulation results presented in Section 4.5.4. For the sync-first strategy in Figure 4.14, the static slot assignment strategy results in overlapping transmit slots, giving rise to empty sets of observable neighbors $\mathcal{N}_i(t)$ as specified in (4.10). For random transmission, on the other hand, empty sets of observable neighbors $\mathcal{N}_i(t)$ are not persistent, and so the union of the dynamic topologies $\mathcal{G}(t)$ results in a connected graph. Hence,

unlike random transmission, sync-first-n/m violates the convergence conditions of [Mor05], which confirms the simulation results presented in Section 4.5.4.

Delayed Coupling Another difference with the PCO model is that, interactions between nodes in MEMFIS are delayed by one period T, when propagation delays are neglected. Hence the evolution of phase variables (4.8) becomes:

$$\frac{\mathrm{d}\phi_i(t)}{\mathrm{d}t} = \frac{1}{T} + \sum_{j \in \mathcal{N}_i(t)} \big(\phi_i(t) + \Delta\phi\big(\phi_i(t)\big)\big) \cdot \delta(t - \tau_j - T) \,. \tag{4.11}$$

In general, in the presence of transmission delays in the order of the natural oscillation period T, it is not possible to prove convergence and the network becomes potentially unstable [Izh98]. Fortunately, the situation in MEMFIS is different, since transmitter j does not modify its phase during a transmit slot. This means that reference instants are fixed when node j transmits, and therefore $\tau_j = \tau_j + T$. Therefore reactions of nodes in MEMFIS are described by an equivalent PCO model where nodes wait for one period before transmitting a pulse. During this waiting period no pulses from other nodes are acknowledged, i.e. the node stays in refractory for one period T. The evolution of phase variables (4.11) becomes equivalent to (4.8), by replacing $\mathcal{N}_i$ with $\mathcal{N}_i(t)$ in (4.8). Hence the phase deviation variables (4.10) appropriately describe delayed coupling, and thus proves convergence for MEMFIS. Apart from the undesirable effect of deafness leading to missed sync-words which degrade the time to convergence, deafness avoids potential instabilities as reported in [Izh98].

4.4 Synchronization Word Detection

The detection of the common synchronization word is an important issue in MEMFIS, because it provides implicit information on the timing of transmitting nodes, which is used to update the clock of a receiving node.

In a meshed network, each node i has a given set of neighbors $\mathcal{N}_i$ that is, for instance, dependent on the position of the nodes and their relative distance in a random geometric graph. In MEMFIS, the number of transmitting nodes at a given instant further depends on the local MAC policies, and thus changes over time. Therefore the number of observable neighbors at instant t at node i, denoted $\mathcal{N}_i(t)$, is also changing over time. As nodes are initially in an asynchronous state, the transmission of a sync-word occurs when other nodes may be transmitting data, in a similar manner to the example in Figure 4.4, which may mask the sync-word and make its detection impossible.

In the remainder of this section, the sync-word detection is evaluated for two receiver types. In Section 4.4.1, detection is performed by correlating the incoming signal with the known sync-word. A simple channel model is assumed, and the detection probability is shown to be proportional to the sync-word length, which confirms the need for a long synchronization sequence to improve the detection. In Section 4.4.2, detection is performed by correlating the received signal with itself and by searching for a known structure. A multicarrier system is considered, and using a known OFDM

synchronization method designed for link synchronization, it is shown that the sync-word can be reliably detected in a multiuser asynchronous environment and reliably used for slot synchronization.

4.4.1 Cross-Correlation Detector

In a wireless environment, the presence of noise and interference make the detection of pulses as assumed for the synchronization of pulse-coupled oscillators virtually impossible. It is therefore more robust to consider a *sync-word* with finite duration. As the synchronization process relies on identifying the timing and not the source, a single sync-word $\{b[k]\}$ common to all nodes is employed.

Transmission The transmitted sequence $\{x_j[k]\}$ contains K symbols and is composed of the following parts: payload data is placed in $\texttt{data}_{\text{pre}}$ and $\texttt{data}_{\text{post}}$, and the sync-word is placed in $\texttt{sync}$. With a sync-word of length M, there are $K-M$ symbols available for payload data, of which D_{pre} are placed in $\texttt{data}_{\text{pre}}$. The symbol sequence transmitted by node j is in the form:

$$x_j[k] = \begin{cases} a_j[k] & , \quad 0 \leq k \leq D_{\text{pre}}-1\,; \\ b[k-D_{\text{pre}}] & , \quad D_{\text{pre}} \leq k \leq D_{\text{pre}}+M-1\,; \\ a_j[k-M] & , \quad D_{\text{pre}}+M \leq k \leq K-1\,. \end{cases} \tag{4.12}$$

Reception Whenever a node i is not transmitting, the incoming signal $y_i(t)$ is directed to the reception unit, and $y_i(t)$ is scanned for sync-words (see Figure 4.4). The start of a signal from node j is observed by node i at time instant $\tilde{\tau}_{ij} = \tau_j + \nu_{ij}$. The propagation delay is given by $\nu_{ij} = d_{ij}/\mathrm{c}$, where d_{ij} accounts for the distance between nodes i and j, and c is the speed of light.

The reception process is considered for slot n_i, beginning when node i fires at instant $\tau_i[n_i]$, and enters a receive slot, i.e. $s_i(\tau_i) = 0$. Denote $\mathcal{J}_i(t)$ the set of transmitting nodes j whose transmissions are observed by node i within the observation window $\tilde{\tau}_{ij} \in [\tau_i[n_i]-T, \tau_i[n_i+1]]$.

Considering an additive white Gaussian noise (AWGN) channel with distance-dependent pathloss, the reception unit receives the signal:

$$y_i(t) = \sum_{j\in\mathcal{J}_i(t)} \sqrt{d_{ij}^{-\chi} P_{\text{s}}} \cdot x_j\big(t-\tilde{\tau}_{ij}\big) + w_i(t) \tag{4.13}$$

where χ is the path loss exponent, P_{s} is the transmit power considered constant for all nodes, and $w_i(t)$ is a Gaussian distributed random variable with zero mean and variance σ^2. The incoming signal $y_i(t)$ is passed through a matched filter with response $g^*(-t)$ and is sampled at time instants $t = kT_{\text{s}}$. The matched filter outputs the sequence

$$\begin{aligned} y_i[k] = &\quad \textstyle\int_{-T_{\text{m}}}^{T_{\text{m}}} g^*(kT_{\text{s}}-t)\, y_i(t)\, \mathrm{d}t \\ = &\quad \textstyle\sum_{j\in\mathcal{J}_i(t)} A_{ij} \cdot x_j[k-\tilde{\tau}_{ij}/T_{\text{s}}] + w_i[k]\,, \end{aligned} \tag{4.14}$$

where $[-T_{\rm m}, T_{\rm m}]$ is the matched filter integration interval, $A_{ij} = \sqrt{d_{ij}^{-\chi} P_{\rm s}}$ accounts for the received amplitude of $x_j(t)$, and $w_i[k] = \int_0^T g^*(kT_{\rm s} - t)\, w_i(t)\, {\rm d}t$ represents the sampled AWGN term at the correlator output.

Sync-Word Detection The sync-word is detected by cross-correlating $\{y_i[k]\}$ in (4.14) with the known sync-word $\{b[m]\}$. It is assumed that only those nodes in $\mathcal{J}_i(t)$ can be detected by receiving node i whose sync-words are fully contained within the observation window. These are the subset of nodes $j \in \mathcal{J}_i(t)$ whose reference instants commence within the time interval $\tilde{\tau}_{ij} \in \left[\tau_i[n_i] - T_{\rm d,pre}, \tau_i[n_i+1] - T_{\rm Tx} - T_{\rm d,pre}\right]$, denoted by $\mathcal{R}_i(t) \subseteq \mathcal{J}_i(t)$.

The resulting correlation metric of node i yields

$$\begin{aligned} \Lambda_i[k] = & \sum_{m=0}^{M-1} b^*[m] \cdot y_i[k+m] \\ = & \begin{cases} M A_{ij} + \bar{z}_i[k]\,, & k = [\theta_{ij}/T_{\rm s}],\ j \in \mathcal{R}_i(t) \\ z_i[k]\,, & \text{elsewhere}\,. \end{cases} \end{aligned} \tag{4.15}$$

where $[.]$ is the closest integer operator. At time instant $\theta_{ij} = \tilde{\tau}_{ij} + T_{\rm d,pre} + T_{\rm Tx}$, the sync-word from node $j \in \mathcal{R}_i(t)$ is observed at node i. The interference terms $\bar{z}_i$ and z_i account for the cross-correlation of the sync-word $\{b[m]\}$ with noise and data from interfering nodes in $\mathcal{J}_i(t)$, including respectively excluding the signal of node $j \in \mathcal{R}_i(t)$.

The cross-correlation of $\{b[m]\}$ with $\{x_j[m + \theta_{ij}/T_{\rm s}]\}$ is described by a sum of M independent Gaussian random variables with zero mean and unitary variance. As $z_i[k]$ is composed of $M|\mathcal{J}_i(t)|$ Gaussian distributed signal components, $z_i[k]$ is a zero-mean Gaussian variable with variance

$$\sigma_z^2[k] = M\left(\sigma^2 + \sum_{\ell \in \mathcal{J}_i(t)} A_{i\ell}^2\right). \tag{4.16}$$

The variance $\sigma_z^2[k]$ is time varying due to the transmission activities of neighboring nodes $j \in \mathcal{J}_i(t)$ at instant $t = kT_{\rm s}$. Likewise, $\bar{z}_i[k]$ is a zero-mean Gaussian distributed random variable with variance $\bar{\sigma}_z^2[k] = \sigma_z^2[k] - MA_{ij}^2$.

As shown in Figure 4.4, a correlation peak at (4.15) is declared as a received sync-word conditioned on the threshold test $\Lambda_i[k] > \Lambda_i^{[\rm th]}[k]$, where the detection threshold $\Lambda_i^{[\rm th]}[k]$ is set as a function of the normalized threshold λ, given by

$$\lambda = \frac{\Lambda_i^{[\rm th]}[k]}{\sqrt{P_i[k]}}\,. \tag{4.17}$$

The average received power $P_i[k] = \sum_{m=0}^{K-1} |y_i[k+m]|^2$ estimates the variance of the received signal, i.e. $P_i[k] \approx \sigma_z^2[k]$. The appropriate choice of λ is important and is elaborated in the following.

In practice the simple threshold detector $\Lambda_i(t) > \Lambda_i^{[\rm th]}(t)$ may be complemented by a more sophisticated link-level synchronization unit [OM88, GG98] so to achieve a timing accuracy of θ_{ij} within a fraction of the sampling duration $T_{\rm s}$. For linearly modulated signals, symbol synchronization is commonly performed by oversampling and exploiting the cyclostationarity of the transmitted

signal $x_j(t)$ [OM88]. In the context of network synchronization, the received signal $y_i(t)$ in (4.13) closely resembles a signal originating from a single source that propagates through a multipath channel. Symbol synchronization in such a scenario is comprehensively studied in the literature, and techniques based on oversampling and exploiting the cyclostationary properties of (4.13), as e.g. [GG98], are therefore directly applicable to MEMFIS.

Provided that the correlation peak $\Lambda_{ij}^{\star} = \Lambda_i[\theta_{ij}/T_\mathrm{s}]$ exceeds the threshold (4.17), the reference instant of node j is successfully detected, so that $\widehat{\tau}_{ij} = \tilde{\tau}_{ij}$. The set of nodes whose transmissions are successfully detected forms the set $\mathcal{N}_i(t) \subseteq \mathcal{R}_i(t) \subseteq \mathcal{J}_i(t)$. Then the start of the transmit slot from node j is determined as $\widehat{\tau}_{ij} = \theta_{ij} - (T_{\mathrm{d,pre}} + T_\mathrm{Tx})$.

From (4.15) the following conclusions regarding slot synchronization can be drawn:

- The output of the correlation detector (4.15) produces a series of peaks $\{\Lambda_i[\theta_{ij}/T_\mathrm{s}]\}_{j\in\mathcal{N}_i(t)}$ in a similar way to the received pulses in the PCO model.
- A sync-word emitted by node j at instant τ_j results in a correlation peak of receiving node i at θ_{ij}, accumulating to an overall transmission delay of $\theta_{ij} - \tau_j = T_{\mathrm{d,pre}} + T_\mathrm{Tx} + \nu_{ij}$.
- A transmitting node j does not modify its reference instant τ_j during its transmit slot, since node j is not able to listen to sync-words from other nodes.
- Due to signal processing in (4.15), a decoding delay T_dec is encountered, which accounts for the time between the correlation peak's actual occurrence, θ_{ij}, and the instant this peak is detected at $\theta_{ij} + T_\mathrm{dec}$. For a correlation detector it is reasonable to assume that T_dec is constant and equal for all nodes.
- The SINR at the output of the sync-word detector is M times the symbol SINR (4.18), $M\gamma_{ij}$. This SINR boost is equivalent to the processing gain of spread spectrum signals with sequence length M.
- If nodes fire and transmit synchronously, so that $\theta_{ij} = \theta_{i\ell}$, $j \neq \ell$, peaks in (4.15) superimpose constructively.

Symbol Detection If $\tilde{\tau}_{ij}$ is successfully detected, the symbol sequence $\{\widehat{x}_i[k] = y_i[k - \tilde{\tau}_{ij}/T_\mathrm{s}]\}$, $0 \leq k < K$ is passed to the decoder (block DEC in Figure 4.2). According to (4.16), the SINR $\gamma_{ij}[k] = A_{ij}^2/\bar{\sigma}_z^2[k]$ is time varying during one slot. The SINR at instant k where the interference $\bar{\sigma}_z^2[k]$ is maximum is chosen as a lower bound for the effective SINR of a received packet $\{\widehat{x}_j[k]\}$ from node j. With this lower bound of the effective SINR γ_{ij}, a packet is assumed to be received successfully if γ_{ij} exceeds a certain threshold γ_th:

$$\gamma_{ij} = \min_k \gamma_{ij}[k] > \gamma_\mathrm{th} \,. \tag{4.18}$$

False Alarm and Detection Probabilities

The appropriate choice of the threshold $\Lambda_i^{[\mathrm{th}]}[k]$ in (4.6) is important, as $\Lambda_i^{[\mathrm{th}]}[k]$ makes hard decisions whether correlation peaks of $\Lambda_i[k]$ in (4.15) are declared as sync-words or not. The sync-word detector tests two hypotheses:

$\overline{\mathcal{H}}$: no sync-word was transmitted;

$\mathcal{H}$: a sync-word was transmitted.

To reflect a realistic detection process, both false alarms and missed detections for the sync-word detection need to be accounted for.

False Alarm Probability A false alarm occurs if the sync-word detector detects a sync-word although none has been transmitted. It occurs if there is an unwanted peak, caused by a high correlation of noise and/or data with the sync-word. The false alarm probability is formally defined as the probability of choosing $\mathcal{H}$ when $\overline{\mathcal{H}}$ is true. The false alarm probability for one symbol period kT_s yields

$$P_\mathrm{fa} = \Pr\left\{\Lambda_i[k] \geq \Lambda_i^{[\mathrm{th}]}[k] \,|\, \overline{\mathcal{H}}\right\} . \tag{4.19}$$

Under hypothesis $\overline{\mathcal{H}}$, no sync-word was transmitted, and the correlator $\Lambda_i[k]$ in (4.15) thus outputs $z_i[k]$, which is a zero-mean Gaussian random variable with variance $\sigma_z^2[k]$ given by (4.16). In this case the correlator output for receiving node i is equal to $\Lambda_i[k] = z_i[k]$ in (4.15), which is the output when correlating the known sync-word with the sum of ongoing transmissions and noise. Assuming that data symbols and the sync-word are taken from the alphabet $\{-1,+1\}$, then $z_i[k]$ is distributed according to a binomial distribution, which for large M is well approximated by the Gaussian probability density function (pdf) $f_0(x)$ with zero mean and variance $\sigma_z^2[k]$.

The false alarm probability P_fa as defined in (4.19) is the probability that under $\overline{\mathcal{H}}$, the correlator output is superior to a given threshold $\Lambda_i^{[\mathrm{th}]}[k]$. Thus P_fa is determined by the tail of the Gaussian pdf $f_0(x)$:

$$\begin{aligned} P_\mathrm{fa} &= \int_{\Lambda_i^{[\mathrm{th}]}[k]}^{\infty} f_0(x)\mathrm{d}x \\ &= Q\left(\frac{\Lambda_i^{[\mathrm{th}]}[k]}{\sigma_z[k]}\right) . \end{aligned} \tag{4.20}$$

where the Q-function is defined by

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-t^2/2}\mathrm{d}t . \tag{4.21}$$

Assuming that the interference term $\sigma_z[k]$ in (4.16) is perfectly known, the detection threshold (4.17) amounts to $\Lambda_i^{[\mathrm{th}]}[k] = \lambda\sigma_z[k]$. Then the final result for the false alarm probability $P_\mathrm{fa} = Q(\lambda)$ is obtained from (4.17) and (4.20). Notice that the false alarm probability is entirely determined by the normalized threshold λ (4.17).

Given $K = T/T_s$ detection instants per time slot, the total false alarm probability per slot amounts to:

$$P_{\text{fa}}^{[\text{slot}]} = 1 - (1 - P_{\text{fa}})^K \ . \tag{4.22}$$

Table 4.1 gives the normalized detection threshold λ that must be chosen to achieve a desired false alarm probability $P_{\text{fa}}^{[\text{slot}]}$. If λ augments, the system becomes more cautious as less unwanted peaks are acknowledged, expressed by a reduced $P_{\text{fa}}^{[\text{slot}]}$.

$P_{\text{fa}}^{[\text{slot}]}$	10^{-4}	10^{-3}	10^{-2}	10^{-1}
$K = 80$	4.708	4.215	3.661	3.008
$K = 160$	4.848	4.369	3.835	3.212
$K = 320$	4.983	4.518	4.002	3.406
$K = 640$	5.116	4.662	4.163	3.591

Table 4.1: Normalized sync-word detection threshold λ.

Detection Probability A correct detection occurs if a sent sync-word is successfully detected. The detection probability is formally defined as the probability of choosing $\mathcal{H}$ when $\mathcal{H}$ is true:

$$P_{\text{d}} = \Pr\left\{\Lambda_i[k] \geq \Lambda_i^{[\text{th}]}[k] \,|\, \mathcal{H}\right\} \ . \tag{4.23}$$

Under hypothesis $\mathcal{H}$, a sync-word was transmitted by node j, so the correlator output $\Lambda_i[k]$ in (4.15) produces a peak with magnitude MA_{ij} plus the interference term $\bar{z}_i[k]$. Similar to the distribution of $z_i[k]$, the statistics of $\bar{z}_i[k]$ are well approximated by the Gaussian pdf $f_1(x)$. Unlike the pdf $f_0(x)$ for $\overline{\mathcal{H}}$ which is zero mean with variance σ_z^2, the mean of $f_1(x)$ is equal to the correlation peak's magnitude MA_{ij}, while its variance is reduced by the power in the sync-word $\bar{\sigma}_z^2[k] = \sigma_z^2[k] - MA_{ij}^2$. Therefore, under $\mathcal{H}$, the correlator output $\Lambda_i[k]$ in (4.15) is a Gaussian distributed random variable with mean MA_{ij} and variance $\bar{\sigma}_z^2[k]$.

The detection rate is given by the probability that the correlator output $\Lambda_i[k] = MA_{ij} + \bar{z}_i[k]$ exceeds the threshold $\Lambda_i^{[\text{th}]}[k]$, which can be expressed by the Q-function:

$$\begin{aligned} P_{\text{d}} &= \int_{\Lambda_i^{[\text{th}]}[k]}^{\infty} f_1(x)\mathrm{d}x \\ &= Q\left(\frac{\Lambda_i^{[\text{th}]}[k] - MA_{ij}}{\bar{\sigma}_z[k]}\right) \end{aligned} \tag{4.24}$$

where $f_1(x)$ is the pdf of the correlator output $MA_{ij} + \bar{z}_i[k]$ (4.15). Inserting $\Lambda_i^{[\text{th}]}[k] = \lambda\sigma_z[k]$ from (4.17) into the Q-function yields

$$\frac{\Lambda_i^{[\text{th}]}[k] - MA_{ij}}{\bar{\sigma}_z[k]} = \lambda\sqrt{1 + \frac{MA_{ij}^2}{\bar{\sigma}_z^2[k]}} - M\sqrt{\frac{A_{ij}^2}{\bar{\sigma}_z^2[k]}} \ . \tag{4.25}$$

It is convenient to express the detection probability as a function of the SINR $\gamma_{ij}[k] = A_{ij}^2/\bar{\sigma}_z^2[k]$, which results in:

$$P_{\text{d}} = Q\left(\lambda \cdot \sqrt{1 + \gamma_{ij}[k]} - \sqrt{M \cdot \gamma_{ij}[k]}\right) \ . \tag{4.26}$$

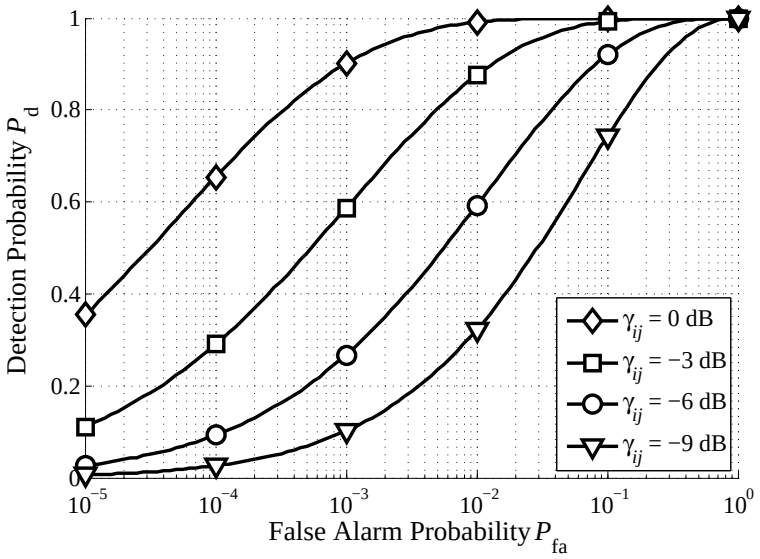


Figure 4.6: Receiver operating characteristic of the cross-correlation receiver for different SINR rates with a sync-word length of $M = 32$.

While increasing the sync-word length M enhances the detection of sync-words, the overhead grows in proportion to M.

A missed detection accounts for the event that a sync-word was transmitted ($\mathcal{H}$ is true) but is not detected at the receiver ($\overline{\mathcal{H}}$ is decided). The missed detection probability is given by $1 - P_{\mathrm{d}}$.

Maximum Likelihood Detector An important task in the design of the sync-word detector is to find a good compromise between a low false alarm rate and a low missed detection rate for a given SINR. Application of the maximum likelihood (ML) criterion leads to the Neyman-Pearson detector [Tre01]. First the normalized detection threshold λ in (4.17) is set so to achieve a desired false alarm probability $P_{\mathrm{fa}}^{[\mathrm{slot}]}$ in (4.22). Second, once λ is identified, the maximum detection probability P_{d} in the ML sense is obtained from (4.26). Figure 4.6 plots the Receiver Operating Characteristic (ROC), i.e. the detection probability as a function of the desired false alarm rate, for a synchronization sequence of length $M = 32$ symbols. A reduction of P_{fa} results in an enlarged detection threshold λ which in turn reduces P_{d}. The appropriate choice for false alarm and detection probabilities for MEMFIS is further studied in Section 4.5.3.

4.4.2 Auto-Correlation Detector

In the following, nodes communicate over an OFDM air interface, such as the wireless LAN standard IEEE 802.11 [IEE99]. In such systems each transmitted packet is typically composed of a preamble and of payload data (see Section 4.6). The synchronization preamble, which is common to all nodes, is needed to detect the start of a packet, adjust the automatic gain control and synchronize the receiver's local oscillator in time and frequency to the transmitter's. For MEMFIS, the

timing information of the preamble is also used for slot synchronization to update the internal clock.

The goal of this section is to verify whether conventional OFDM timing synchronization schemes, devised for synchronization of an essentially isolated link, are applicable to decentralized slot synchronization protocol. To this end, the method proposed in [MBL03] is employed; this method offers a robust timing estimator designed by searching for the correlation peak among the repetitive parts of the synchronization preamble, i.e. by auto-correlation of the signal. In spite of its effectiveness in a scenario where only one transmitter is active, this approach does not provide any guarantee to detect the arrival of a synchronization preamble in a multiuser asynchronous network. The main impairment is represented by the interference from other transmitting nodes, in the form of payload data and misaligned synchronization words. The interference tends to mask synchronization peaks and may also give rise to spurious peaks that do not correspond to any received preambles. This may disrupt the slot synchronization algorithm, as these spurious peaks may be misinterpreted as detected preambles from other nodes.

The algorithm is evaluated in a similar manner to the cross-correlation receiver in Section 4.4.1, i.e. in terms of false alarm and detection rates. A different channel model is considered, a multitap fading channel in this case, and the impact of overlapping interval of sync-words clarified.

Detection of the Synchronization Word in a Multicarrier System

In this section, the system model is presented and two OFDM synchronization algorithms are reviewed. The synchronization unit scans the received signal stream in order to detect synchronization preambles that mark the start of a new packet. For MEMFIS, the synchronization unit cannot rely on any *a priori* information, such as the timing of previously received packets, because transmitting nodes may change their reference instants over time.

System Model Without loss of generality, the considered receiving node i receives signals from $\mathcal{J}_i$ neighboring nodes. Each frame is composed of N_B data blocks. The considered baseband-equivalent discrete-time signal model has a sampling period $T_\mathrm{s} = T_\mathrm{B}/N_\mathrm{s}$, where N_s is the number of available subcarriers and $1/T_\mathrm{B}$ the subcarrier spacing. The time-domain samples of the jth transmitter, $1 \leq j \leq \mathcal{J}_i$, during the pth OFDM block are expressed by:

$$x_{j,p}[k] = \frac{1}{\sqrt{N_\mathrm{s}}} \sum_{n=0}^{N_\mathrm{s}-1} a_{j,p}[n] \cdot \exp\left(\frac{\mathrm{j}2\pi n k}{N_\mathrm{s}}\right),\ -N_\mathrm{g} \leq k \leq N_\mathrm{s} - 1 \tag{4.27}$$

where N_g is the length of the cyclic prefix (CP) expressed in sampling periods while $a_{j,p}[n]$ is the modulated symbol over the nth subcarrier. The signal transmitted by the jth transmitter is the concatenation of N_B adjacent blocks and is given by:

$$x_j[k] = \sum_{p=0}^{N_\mathrm{B}-1} x_{j,p}[k - pN_\mathrm{b}] \tag{4.28}$$

where $N_\mathrm{b} = N_\mathrm{s} + N_\mathrm{g}$ is the block length including the CP.

Each stream $x_j[k]$ propagates through a multipath channel characterized by an impulse response $\mathbf{h}_{ij} = [h_{ij}[0], h_{ij}[1], \ldots, h_{ij}[L-1]]^{\mathrm{T}}$ of length L (in sampling periods), and arrives at the considered terminal with some synchronization mismatch relative to the local time and frequency scales. The timing offset is denoted ψ_{ij}, i.e. the timing misalignment of the frame start to the jth user with regards to the receiving node i, expressed in sampling periods, while ϑ_{ij} is the frequency offset normalized to the subcarrier spacing. After baseband conversion and sampling, the signal stream at the receiving node i is in the form

$$y_i[k] = \sum_{j=1}^{\mathcal{J}_i} y_{ij}[k - \psi_{ij}] + w_i[k] \tag{4.29}$$

where $w_i[k]$ represents complex-valued additive white Gaussian noise (AWGN) with variance σ^2 while $y_{ij}[k]$ is the signal from the jth transmitter:

$$y_{ij}[k] = \exp\left(\frac{\mathrm{j}2\pi\vartheta_{ij}k}{N_{\mathrm{s}}}\right) \sum_{\ell=0}^{L-1} h_{ij}[\ell] \cdot x_j[k-\ell]\ . \tag{4.30}$$

Timing Metric A desirable property of a timing estimation scheme is its robustness to frequency offsets. If the training block exhibits a repetitive structure in the time-domain, then a robust timing estimator can be designed by searching for the correlation peak among the repetitive parts. Solutions in this sense have been proposed in [SC97, MBL03], where the synchronization preamble is composed of several identical parts with possible sign inversions. In the following, the synchronization word is a training block that is the same for all nodes, and is composed of Q repeated segments $\mathbf{s}_q$, $0 \leq q \leq Q-1$, in the time-domain. Each segment has P samples and is expressed by:

$$\mathbf{s}_q = b_q \cdot \mathbf{s} \tag{4.31}$$

where $\mathbf{s}$ is a P-dimensional vector independent of the segment index q, while $b_q \in \{\pm 1\}$ are bipolar symbols which determine the training pattern $\mathbf{b} = [b_0, b_1, \ldots, b_{Q-1}]^{\mathrm{T}}$. The training block is preceded by a cyclic prefix of length N_{g} and has a total of $M = N_{\mathrm{g}} + PQ$ samples. Hence, the number of samples in each frame is found to be $N_{\mathrm{F}} = M + N_{\mathrm{B}} \cdot N_{\mathrm{b}}$, and the total duration of the frame is equal to $T = N_{\mathrm{F}} T_{\mathrm{B}}$.

One of the first OFDM synchronization algorithms was presented by Schmidl and Cox in [SC97]. The synchronization word is composed of two identical halves, i.e. $Q = 2$, and detection is performed by correlating one half with the other. This produces a large synchronization peak, and enables simple detection of transmitted sync-words. Figure 4.7 shows an example of the produced peak for one transmitter.

More recently, Bhargava *et al.* presented in [MBL03] a generalized version of the Schmidl and Cox algorithm. The sync-word is composed of more repetitions, which can be sign inverted. The sign inversion pattern is performed by an exhaustive search among available patterns. For $Q = 16$,

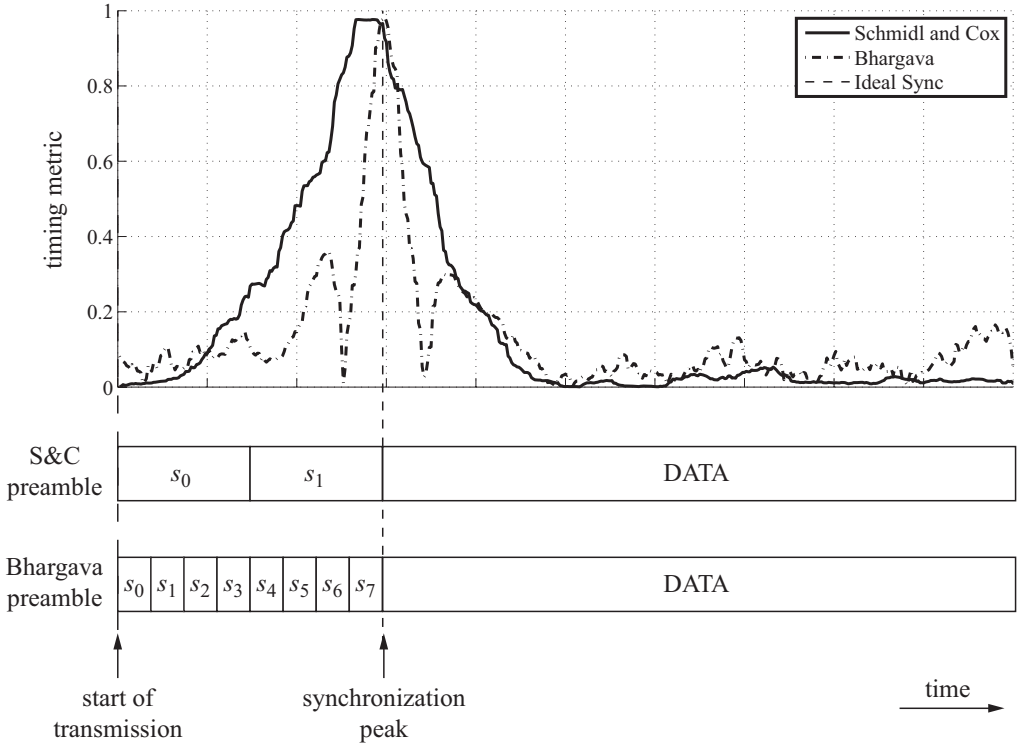


Figure 4.7: Behavior of the Schmidl and Cox and the Bhargava synchronization algorithms for one transmitter.

the best training block is formed using the following pattern:

$$\mathbf{b} = [1,-1,-1,1,1,1,-1,-1,1,-1,1,1,-1,1,-1,-1]^{\mathrm{T}} . \tag{4.32}$$

Detection of this synchronization pattern produces a sharp and narrow peak, as depicted in Figure 4.7. The detector also produces side-lobes, which may affect the detection.

These synchronization algorithms are typically evaluated in a single-user environment to detect the start of transmission of the transmitting node, and are evaluated with regards to the impact of the noise level and channel model. For MEMFIS, a further requirement is the need to distinguish several simultaneous transmissions of the sync-word that are transmitted asynchronously. To this end, the narrow peak of the Bhargava algorithm is more appropriate then the Schmidl and Cox algorithm when synchronization words overlap. This case is shown in Figure 4.8 for two transmitters.

In Figure 4.8, the Schmidl and Cox detector does not produce two easily identifiable peaks, but one large peak with a plateau with a width equal to the user separation ψ_{ij}. On the other hand, the Bhargava algorithm produces two peaks with maximums indicating the end of each synchronization word. The superposition of the two sync-words also increases the height of the side-lobes. The impact of these on the detection is evaluated in Section 4.4.2.

The algorithm of [MBL03] is thus preferred to other OFDM synchronization algorithms, because its timing metric exhibits a high and narrow peak when a synchronization preamble is present.

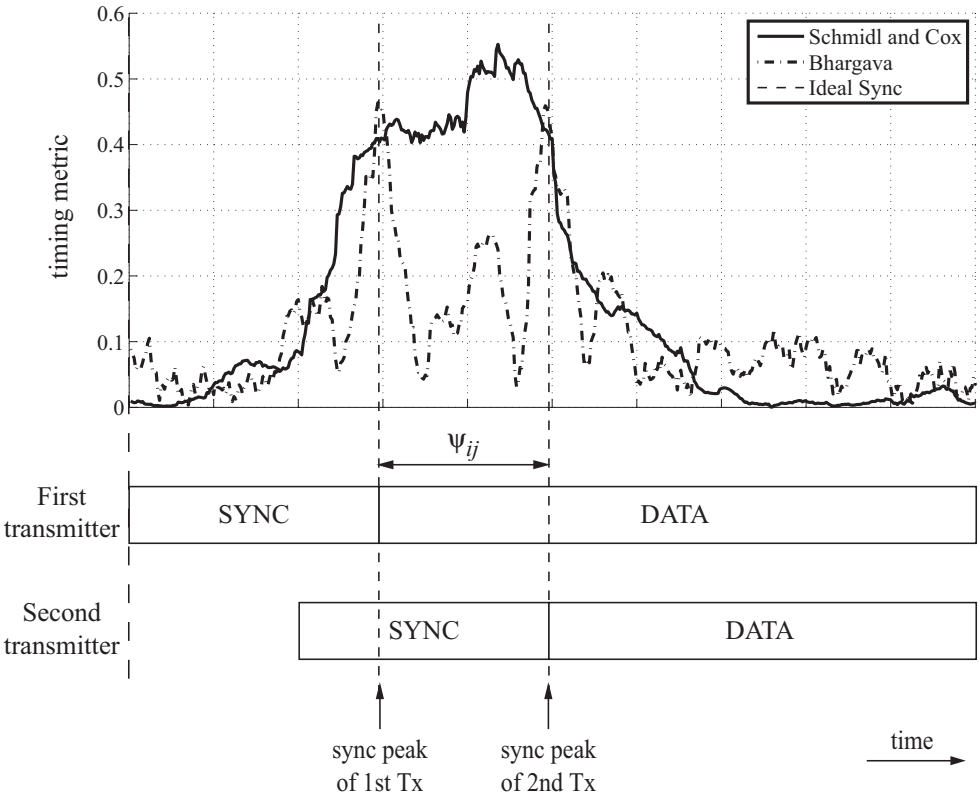


Figure 4.8: Behavior of the Schmidl and Cox and Bhargava synchronization algorithms for two transmitters with overlapping synchronization words.

This way, when two preambles are relatively close, which is typical for a multiuser asynchronous scenario, the metric should exhibit two peaks. This section describes the generation of the timing metric of [MBL03].

Let $k = 0$ denote the start of a frame in the time scale of the considered terminal, which observes the received signal $y_i[k]$ expressed by (4.29) and (4.30). The timing metric $\Lambda_i[k]$ is computed for detecting the synchronization word. It is equal to [MBL03]:

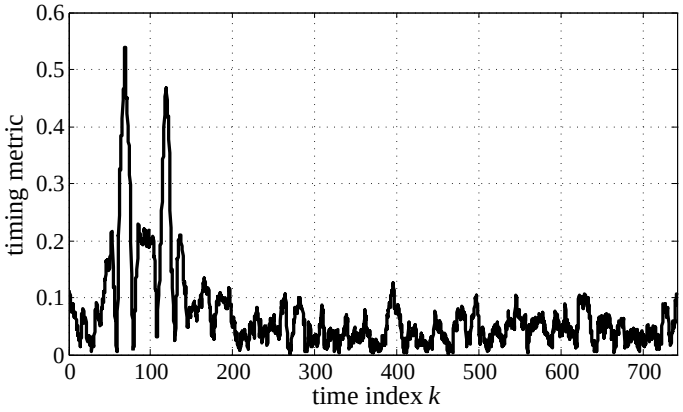
$$\Lambda_i[k] = \frac{Q}{Q-1} \cdot \frac{|\mu_i[k]|}{E_i[k]} \text{ for } 0 \leq k \leq N_F - 1 \tag{4.33}$$

where $\mu_i[k]$ and $E_i[k]$ are defined as:

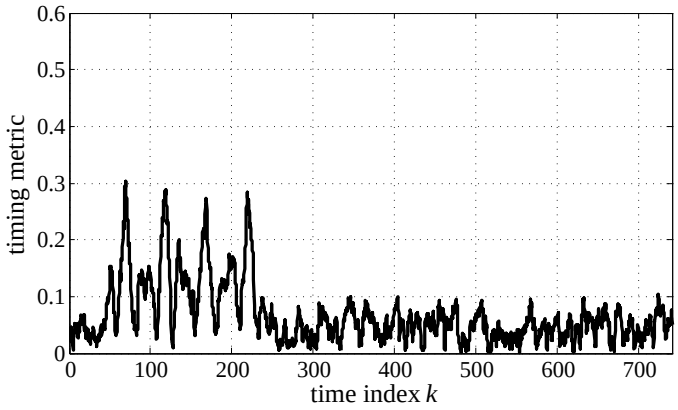
$$\begin{aligned} \mu_i[k] \quad &= \quad \textstyle\sum_{q=0}^{Q-2} b_q \cdot b_{q+1} \cdot \rho\,[k+qP] \\ &\text{with} \quad \rho[\ell] = \sum_{p=0}^{P-1} y_i[\ell+p+P] \cdot y^*[\ell+p] \end{aligned} \quad , \tag{4.34}$$

and

$$E_i[k] = \sum_{q=0}^{Q-1} \sum_{p=0}^{P-1} |y_i\,(k+qP+p+jN_F)|^2 \tag{4.35}$$



(a) $\mathcal{J}_i = 2$



(b) $\mathcal{J}_i = 4$

Figure 4.9: Timing metric examples.

where the superscript * denotes the complex conjugate and $|\cdot|$ denotes the absolute value of the enclosed quantity. Figure 4.9 illustrates a realization of $\Lambda_i[k]$ in the case of two and four active nodes with delays $\psi_{ij} = 160 + 100 \times j$ samples, for $0 \leq j \leq 3$.

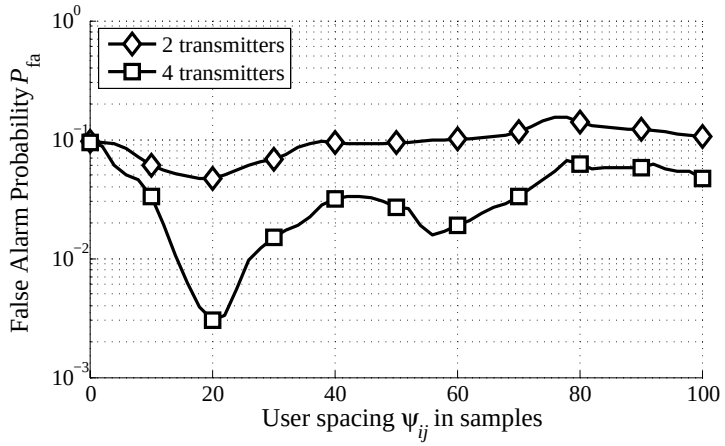
Each segment is made of $P = 16$ samples while the CP has length $N_g = 32$. This corresponds to a training block with overall duration $M = 288$. The number of subcarriers per OFDM symbol is set to $N_s = 128$. The channel responses $\mathbf{h}$ are modeled as statistically independent Gaussian random vectors with zero-mean and an exponentially decaying power delay profile. The channels have length $L = 8$ and the average received energy is the same for each user. For simplicity, the frequency offsets are set to zero.

As expected, in Figure 4.9 the timing metric exhibits a peak whenever a synchronization preamble of a given user arrives at the receiving terminal. However the presence of data blocks from the other nodes interferes with the synchronization preamble, which tends to mask the synchronization peaks and also gives rise to spurious peaks. This is especially visible for $\mathcal{J}_i = 4$ where the spurious peak from the side lobes are almost as high as the synchronization peaks.

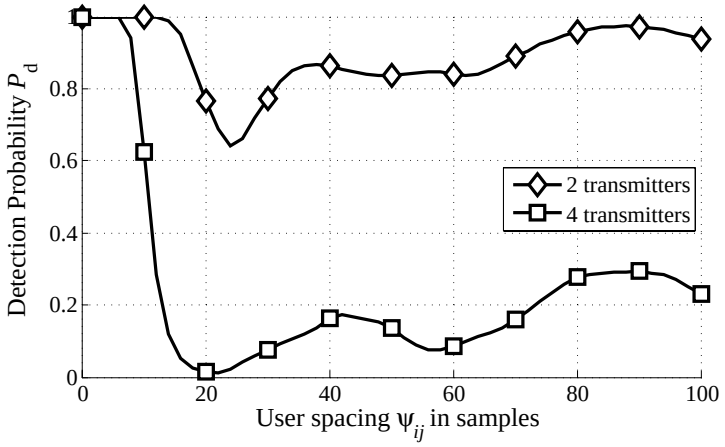
In the following the performance of the timing metric given in (4.33–4.35) is investigated and compared to the cross-correlation receiver presented in Section 4.4.1. The timing metric $\Lambda_i[k]$ is computed at each sampling instant kT_s using the last PQ received samples as indicated in (4.33–4.35). The arrival of a synchronization preamble is then declared as soon as $\Lambda_i[k]$ overcomes a suitable threshold:

$$\Lambda_i[k] \geq \lambda \,. \tag{4.36}$$

The choice of an appropriate threshold λ is important, as λ makes hard decisions whether correlation peaks of $\Lambda_i[k]$ in (4.33) are declared as a valid synchronization preamble or not.



(a) False alarm rate



(b) Detection rate

Figure 4.10: False alarm and detection rates of the cross-correlation receiver as the inter-user spacing ψ_{ij} varies.

Comparison with the Auto-Correlation Detector

The performance of the auto-correlation detector is evaluated through false alarm and detection rates, in a similar manner to the cross-correlation detector previously. The synchronization word detector tests two hypotheses: $\overline{\mathcal{H}}$, no signal or data was transmitted, and $\mathcal{H}$, a synchronization word was transmitted.

A key difference between the cross-correlation and auto-correlation receivers concerns the width of the synchronization peaks. In the former case, the peak is exactly one sample wide, and no side lobes are present. In the latter case, the side lobes may interfere with the synchronization peaks if the inter-user spacing becomes too small. Figure 4.10 plots the variation of the false alarm and detection rates for $\mathcal{J}_i = 2$ and $\mathcal{J}_i = 4$ asynchronous users as the inter-user spacing ψ_{ij} varies.

In Figure 4.10, the detection threshold λ is set according to the Neyman-Pearson criterion for two simultaneous transmitters, i.e. $\mathcal{J}_i = 2$, with a spacing of $\psi_{ij} = 100$ samples and a false alarm rate of 10^{-1}. At this point, the detection rate is equal to $P_{\mathrm{d}} = 0.9$. Reducing the inter-user spacing does not have a major impact on the false alarm rate, which remains around $P_{\mathrm{fa}} \approx 10^{-1}$. The detection rate also does not change until the spacing drops below $\psi_{ij} = 30$ samples. For a spacing of around $\psi_{ij} \approx 20$ samples, the impact of side lobes is observed; they interfere with neighboring peaks, and abruptly lower P_{d}. Further decreasing the inter-user spacing causes the synchronization peaks to merge, and results in a maximal detection probability of 1. As the normalized detection threshold is set for $\mathcal{J}_i = 2$ simultaneous users, increasing $\mathcal{J}_i$ to 4 causes the false alarm and detection rates to drop. This can be seen in Figure 4.9, where the amplitude of the metric is reduced, both for the synchronization peaks and the noisy part of the metric. Varying the inter-user spacing in this case has a similar effect to the two user case.

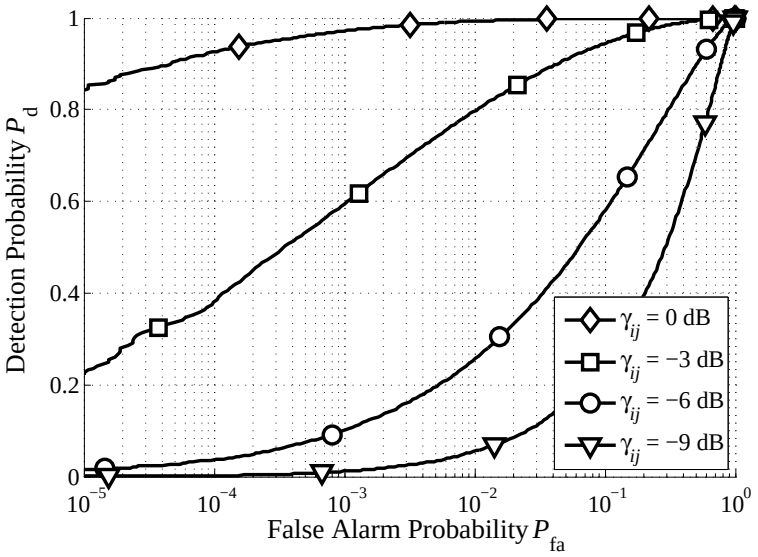


Figure 4.11: Receiver operating characteristic of the auto-correlation receiver for different SINR rates with a sync-word of $Q = 8$ repetitions of segments with $P = 16$ samples each.

To further examine the difference between the two receiver types, Figure 4.11 plots the ROC for a synchronization sequence of $Q = 8$ repetitions and a channel length of $L = 12$ taps with an exponentially decaying profile. Comparing the ROC of this receiver to the ROC of the cross-correlation receiver in Figure 4.6, it can be seen that both have very similar performances. The auto-correlation receiver performs better at higher SINR values, but the detection probability decreases more rapidly than the cross-correlation receiver as the SINR level decreases. Therefore, although their performance also depends on the channel conditions and the synchronization word length, the detection capabilities of these two receivers have similar impact on MEMFIS. The following section assumes the cross-correlation receiver is utilized.

4.5 Performance Evaluation

4.5.1 Modeling Assumptions and Simulation Setup

Network Topology

The network topology is modeled as a random geometric graph $\mathcal{G}$ consisting of a set of N nodes denoted by $\mathcal{V}$ and a set of links denoted by $\mathcal{E}$. The existence of a link between each node pair is determined by the following *path loss model*. Each node transmits with the same power P_s. A node i receives the transmission of a node j at a distance d_{ij} with power $P_\mathrm{s} d_{ij}^{-\chi}$, where χ is the path loss exponent. Two nodes are connected by a link if a minimum signal-to-noise ratio γ_th is maintained between them, i.e. if

$$\frac{P_\mathrm{s} d_{ij}^{-\chi}}{\sigma^2} > \gamma_\mathrm{th} \,. \tag{4.37}$$

Simulations are performed using the following parameters, which are typical for an indoor scenario [Rap01]: $P_\mathrm{s} = 0\,\mathrm{dBm}$, $\sigma^2 = -93\,\mathrm{dBm}$, $\chi = 4$, $\gamma_\mathrm{th} = 5\,\mathrm{dB}$. This translates to a maximum distance between connected nodes of

$$d_\mathrm{max} = 10^{\left(\frac{P_\mathrm{s}}{\mathrm{dBm}} - \frac{\sigma^2}{\mathrm{dBm}} - \frac{\gamma_\mathrm{th}}{\mathrm{dB}}\right)/(10\chi)} = 158\,\mathrm{m} \,. \tag{4.38}$$

Networks are characterized by their algebraic connectivity (see Section 2.4). In order to generate a random topology, all N nodes are placed on a square area using a uniform random distribution. For a given N, the algebraic connectivity κ is varied by changing the area size.

Data Traffic

The arrival of packets is modeled by a Poisson process with arrival rate λ_pkt packets per time slot in the entire network. Once a packet arrives, it is assigned to a randomly chosen node in the network. The MAC schedules the transmission of a packet at the beginning of the next slot. The scheduling of packets appears as a random process to the synchronization unit.

Slot Synchronization

The coupling parameters of the synchronization algorithm are set to $\alpha = 2.00$ and $\beta = 0.01$. The sync-word in MEMFIS is chosen to be a pseudo-random sequence of length $M = 32$ symbols. One slot further contains 288 payload symbols, so that a slot has $K = 320$ symbols in total. The normalized detection threshold λ is set to a particular value so that a desired false alarm rate per slot of $P_{\text{fa}}^{[\text{slot}]}$ is achieved (default value: $P_{\text{fa}}^{[\text{slot}]} = 10^{-3}$). The initial time offsets $t_i(0)$ in (3.2) are taken from a uniform distribution over $t_i(0)/T = [0,1]$.

Performance Metric and Simulation Technique

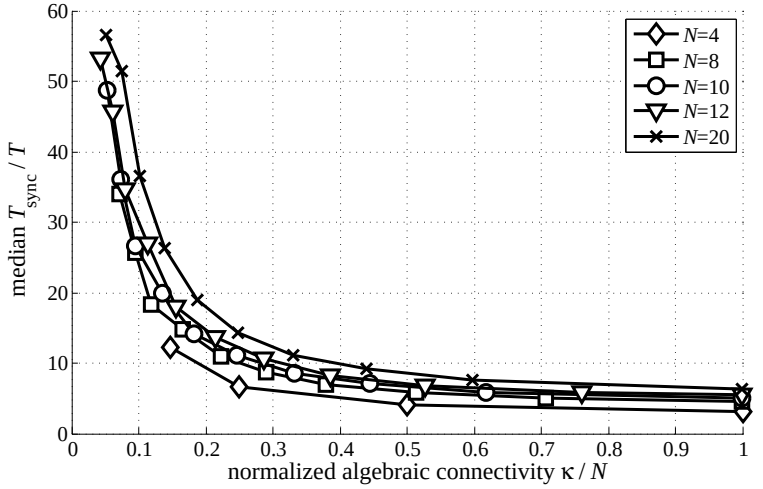
As the primary performance measure, simulations evaluate the time to synchrony T_{sync} normalized by the time slot duration T. Investigations are conducted by measuring the time taken for a network to synchronize starting from a random situation. The cumulative distribution function of T_{sync} and its median value are computed over $5,000$ sets of initial conditions. Simulations are performed using a self-written, MATLAB-based tool that implements the PCO model, MEMFIS, and alternative slot synchronization algorithms and the above modeling assumptions. Unless otherwise stated, simulations use the standard parameters shown in Table 4.2.

4.5.2 Comparison between MEMFIS and the PCO Model

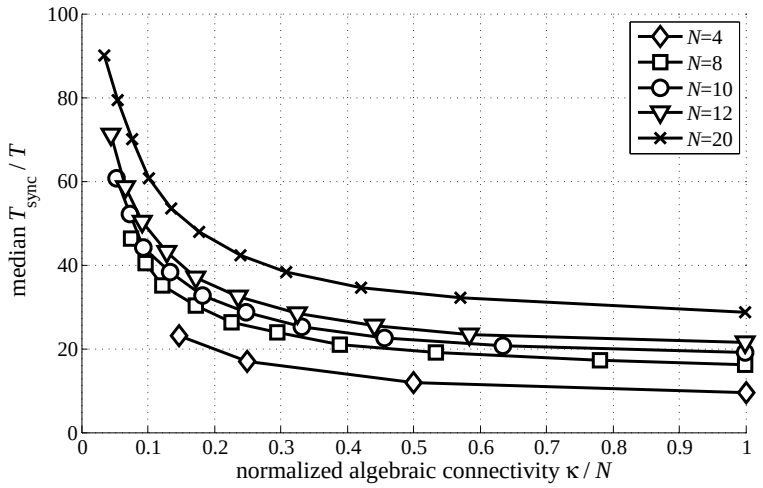
Figure 4.12 plots the resulting curves for both the PCO model and MEMFIS, and analyzes the median value of T_{sync}/T as a function of the algebraic connectivity normalized by the network

Network topology and data traffic		
Transmit power	P_{s}	0 dBm
Noise Power	σ^2	-93 dBm
Path Loss Exponent	χ	4
Minimum SINR	γ_{th}	5 dB
Number of Nodes	N	8
Algebraic Connectivity	κ	1.00
Mean Packet Arrivals	λ_{pkt}	1 pkt/slot
Slot Synchronization		
Coupling Parameters	α	2.00
	β	0.01
Refractory Period	T_{refr}	$0.25\,T$
Packet Length	K	320
Sync-word Length	M	32
	T_{Tx}/T	0.10
Placement of Sync-word	D_{pre}	208
False Alarm Rate per Slot	$P_{\text{fa}}^{[\text{slot}]}$	10^{-3}

Table 4.2: Default simulation parameters.



(a) PCO model



(b) MEMFIS

Figure 4.12: Median time to synchrony as the algebraic connectivity increases.

size κ/N. In both cases, the time to synchrony decreases exponentially as the network gets denser. This result verifies the study by Lucarelli and Wang [LW04], which shows that the time to convergence decreases exponentially with a rate characterized by the product of the weak coupling value ε (see Section 4.3.2), the natural oscillation frequency $1/T$, and the algebraic connectivity κ. Furthermore the time to synchrony for MEMFIS is higher than that of the PCO model, the reason being MEMFIS nodes miss sync-words whilst transmitting, and the coupling is delayed by one period. However, unlike MEMFIS the PCO model assumes that nodes are always able to listen, implying infinitely short pulses and that no payload data may be transmitted.

4.5.3 Robustness Regarding Erroneously Detected Sync-Words

The influence of false alarms and missed detections on MEMFIS is investigated in Figure 4.13 (see Section 4.4). The detection and false alarm probabilities are taken from the receiver operating characteristic in Figure 4.6.

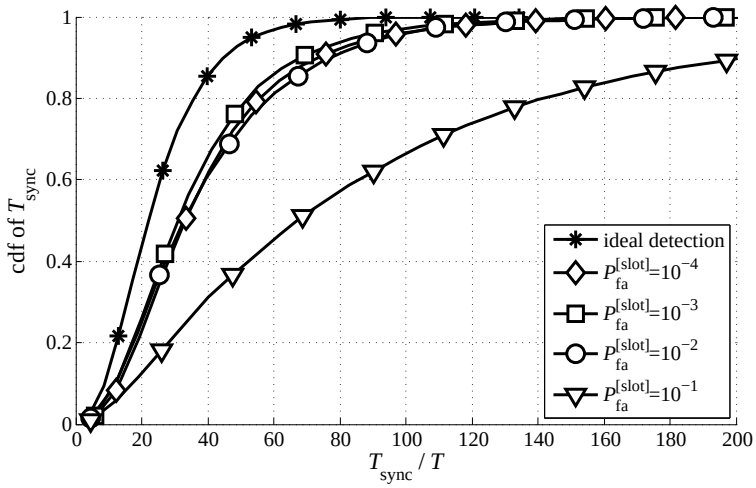


Figure 4.13: Cdf of normalized time to synchrony T_{sync}. Influence of false alarms and missed detections.

From the simulation results in Figure 4.13 the system always converges from any initial misalignment if the detection threshold is correctly chosen. Nevertheless the time to synchrony is affected by spurious peaks, and false alarms are significantly more harmful for slot synchronization than missed detections. Whilst the synchronization of coupled oscillators is sensitive to spurious peaks, it exhibits high robustness when missing some synchronization peaks. Thus, when lowering the threshold λ, the growing false alarm rate P_{fa} implies more spurious peaks, which degrades the time to synchrony. On the other hand, having a too cautious system, i.e. $P_{\text{fa}} \geq 10^{-4}$, also degrades the performance, because too many synchronization are discarded, but the impact is less severe than choosing a low detection threshold. In summary it is thus important to keep the false alarm rate per slot below $P_{\text{fa}}^{[\text{slot}]} \leq 10^{-2}$ to ensure the time to synchrony is not affected. Although this increases the number of discarded synchronization peaks, falsely detected peaks disrupt more heavily the slot synchronization scheme.

4.5.4 Impact of MAC Strategy

To assess the behavior of MEMFIS with respect to the scheduling policy, two MAC strategies are considered. In the "random transmission" strategy, transmissions are scheduled as packets arrive following a Poisson distribution. This is the scheduling policy used in the previous plots. In the "sync-first-n/m" strategy, the objective is to maximize the number of transmitted sync-words in order to reach synchrony as quickly as possible. Nodes transmit for n consecutive time slots and

listen for $m-n$ consecutive time slots in a periodic fashion regardless of the flow of incoming data. Figure 4.14 plots the resulting cdf of the normalized time to synchrony T_{sync}/T.

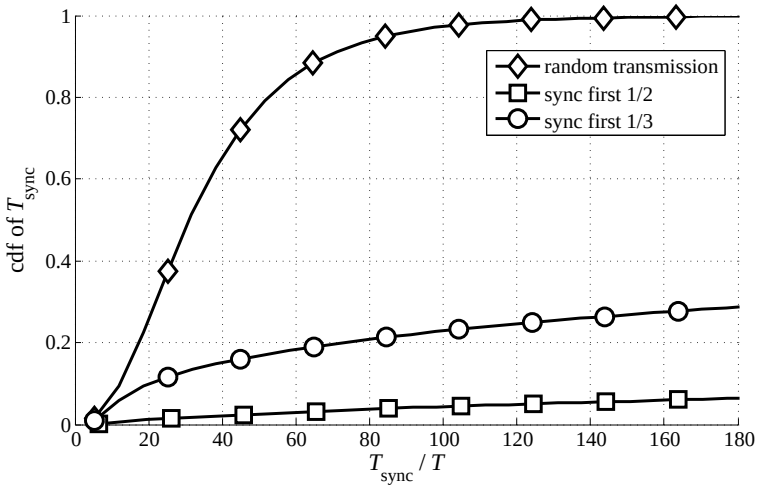


Figure 4.14: Cdf of normalized time to synchrony T_{sync}. Influence of the MAC protocol.

Somewhat surprisingly, synchronization emerges faster for the random transmission strategy, and for any initial condition, convergence to synchrony is accomplished. While increasing the listening time for the sync-first strategy from $n/m = 1/2$ to $1/3$ improves the synchrony rate, for $n/m = 1/3$ in less than 30% of initial conditions the network synchronizes within 180 periods. The poor performance of the sync-first-n/m strategy is explained by the deafness of transmitting nodes. Deafness between nodes follows from the fact that transmitting nodes cannot listen, which means that nodes whose transmit slots mutually overlap are effectively decoupled. For sync-first-n/m, nodes follow a periodic pattern, and if the transmit slots of two nodes overlap, deafness is persistent over time, unless their reference instants τ_i are modified by a third node. If there is no node that transmits while the other nodes are listening, a steady state is reached and the network will never synchronize. With random transmissions, where no predetermined transmit-receive pattern exists, overlapping transmit slots are not persistent. This effectively ensures that failure to synchronize due to deafness is mitigated.

Thus, the fact that nodes listen and transmit randomly fulfills a double requirement. First, it enables a non-fully-meshed network to reach synchrony independently of initial conditions. Second, it imposes less constraints on the MAC layer, so to decide whether to receive or transmit packets.

Further investigating the influence of the MAC strategy, Figure 4.15 plots the resulting cdf of normalized time to synchrony T_{sync}/T for various data traffic densities λ_{pkt}.

Data traffic is closely linked to the emergence of synchronization in MEMFIS. When the traffic is too low, i.e. $\lambda_{\text{pkt}} \leq 1$ packet per slot, synchronization gradually emerges as λ_{pkt} and the number of exchanged packets increase. The time to synchrony is optimal for a mean packet rate of $\lambda_{\text{pkt}} = 2$ packets per slot; this optimal rate depends on the network topology and the ROC of receivers. Further increasing the data traffic does not increase the time to synchrony, which decreases for high

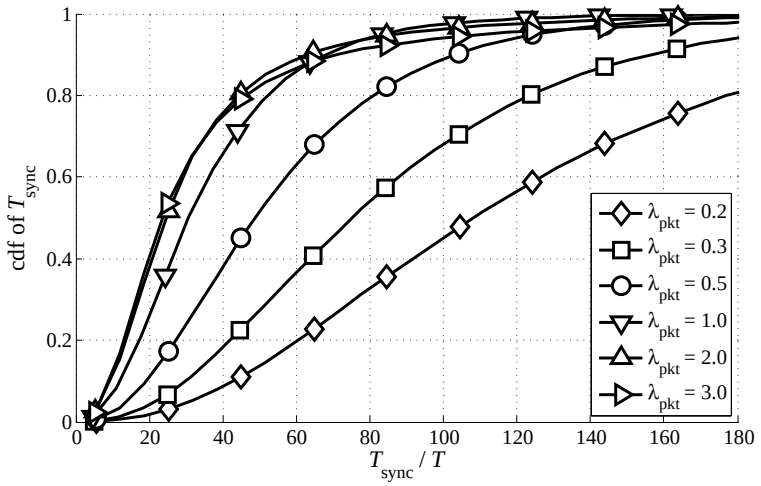


Figure 4.15: Cdf of normalized time to synchrony T_{sync}. Influence of the data traffic density.

traffic rates. As the traffic rate increases, the average SINR at receiving decreases, which makes the detection more difficult, and the increasing number of transmitting nodes also causes sync-words to be missed because of deafness, i.e. half-duplex transmissions.

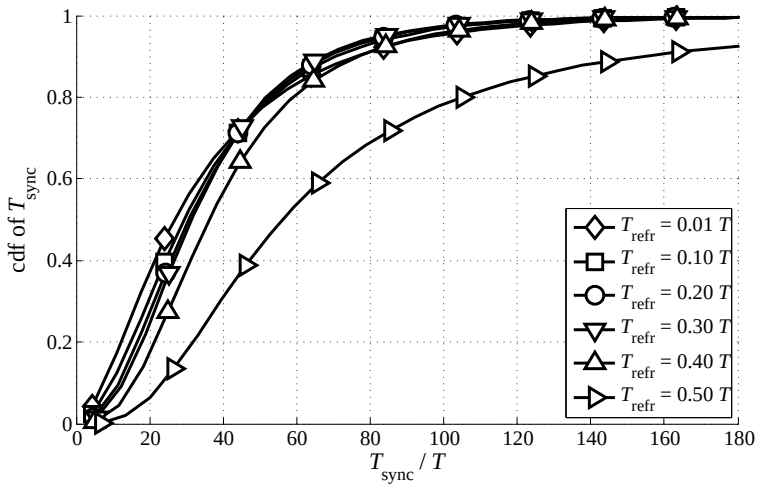
4.5.5 Impact of the Refractory Period

The duration of the refractory period T_{refr} determines the proportion of received sync-words that are discarded after firing. It is an interesting parameter as it dictates whether hearing more sync-words is beneficial in terms of convergence and stability, or if, on the contrary, not reacting to all received transmissions decreases the time to synchrony. Figure 4.16 plots simulation results for various values of the refractory period duration T_{refr}.

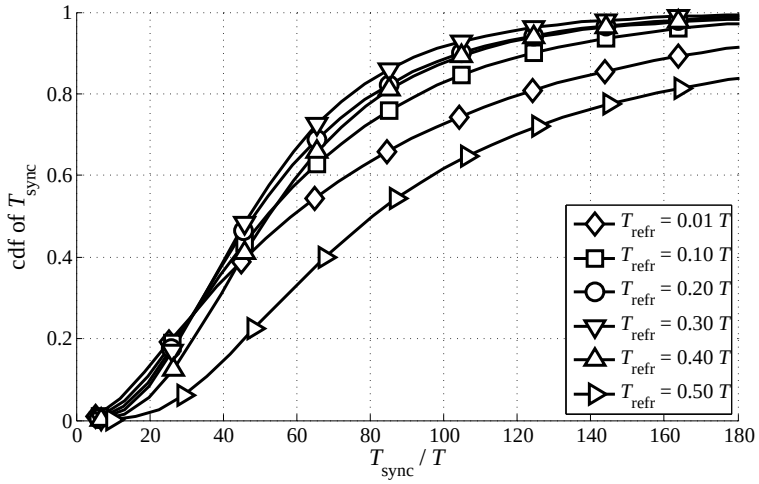
The impact of T_{refr} is evaluated for two network connectivities. In Figure 4.16(a) the algebraic connectivity is set to its default value of $\kappa = 1$. In this case the refractory period has little impact on the time to convergence, and within the range $T_{\text{refr}}/T \in [0.01, 0.40]$, the system converges with approximately the same time. For a high refractory duration of $T_{\text{refr}} = 0.50T$ the system fails to converge rapidly because the listening time is too short. The behavior of the system is very different when lowering the connectivity of the network to $\kappa = 0.5$ in Figure 4.16(b). A certain refractory duration is necessary to achieve fast convergence. If T_{refr} is too low, the network is more unstable than for higher values because nodes are subject to too many conflicting sync-word messages. The optimal performance is obtained for a refractory duration of $T_{\text{refr}} = 0.30T$.

4.5.6 Average Throughput over Time

Slot synchronization offers a key advantage of a higher throughput. The average throughput is here defined as the percentage of sent packets that are successfully received. Increasing the throughput is one of the key incentives for synchronizing a network (see Section 1.2.3). A packet is assumed to be



(a) Algebraic connectivity $\kappa = 1.0$



(b) Algebraic connectivity $\kappa = 0.5$

Figure 4.16: Cdf of normalized time to synchrony T_{sync}. Impact of the refractory duration T_{refr}.

received successfully if its received SINR (4.18) exceeds a certain threshold $\gamma_{ij} > \gamma_{\text{th}}$, which is set to $\gamma_{\text{th}} = 5\,\text{dB}$. Figure 4.17 plots the average throughput over time for different network topologies and a constant mean arrival rate of $\lambda_{\text{pkt}} = 1$ packet per slot. The MEMFIS slot synchronization process is activated after 25 slots.

Before $t = 25\,T$, the average throughput stabilizes around the asynchronous throughput rate. Sparse networks have a higher throughput than dense networks, as the latter have more neighbors, resulting in higher interference. When MEMFIS synchronization is switched on at time instant $t = 25\,T$, it gradually synchronizes the network as packets are transmitted, and the medium access changes seamlessly from unslotted to slotted ALOHA. It is seen that even if the network is only

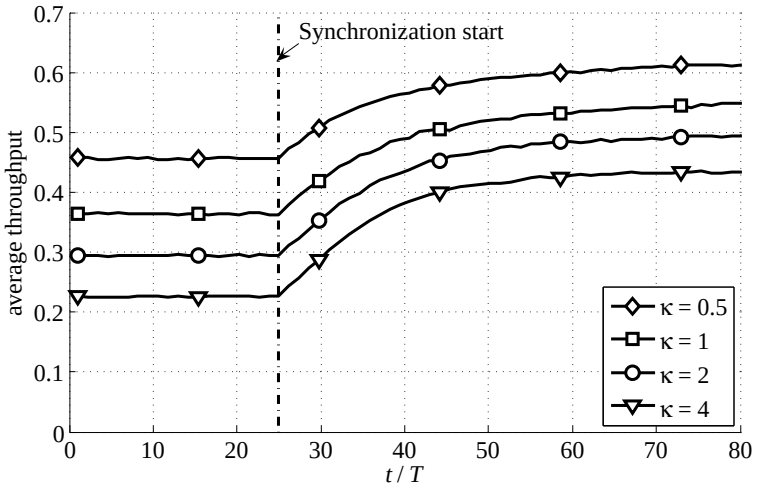


Figure 4.17: Achieved average throughput over time for different algebraic connectivities and for a mean arrival rate of $\lambda_{\text{pkt}} = 1$ packet per slot.

partly synchronized, significant gains in throughput are achieved, and within 25 time slots a stable throughput is almost reached. Similar results are reported in [ERLH02] for the throughput of partially synchronized vehicular networks. The maximum throughput is reached more quickly in denser topologies, as nodes require less time to synchronize (see Figure 4.12).

4.5.7 Comparison with Power-Weighted Average Synchronization

A competing approach for mutual slot synchronization is the Power-Weighted Average Synchronization (PWASync) algorithm. It was initially proposed for inter-base station synchronization in cellular systems [AAK91] and was later extended to the dynamic environment of inter-vehicle communication [SN99, ERLH02]. The application of PWASync in [ERLH02] fits similar constraints as MEMFIS: long sync-words are considered instead of pulses; these are time-multiplexed with data to form packets. An approach similar to PWASync is also presented in [HS06], where distributed synchronization in the case of very high node density is studied.

The basic concept of PWASync is as follows. In slot n node i monitors the detected reference instants $\widehat{\tau}_{ij}[n]$ and the received power level of node j, and determines the power-weighted average timing error [SN99]:

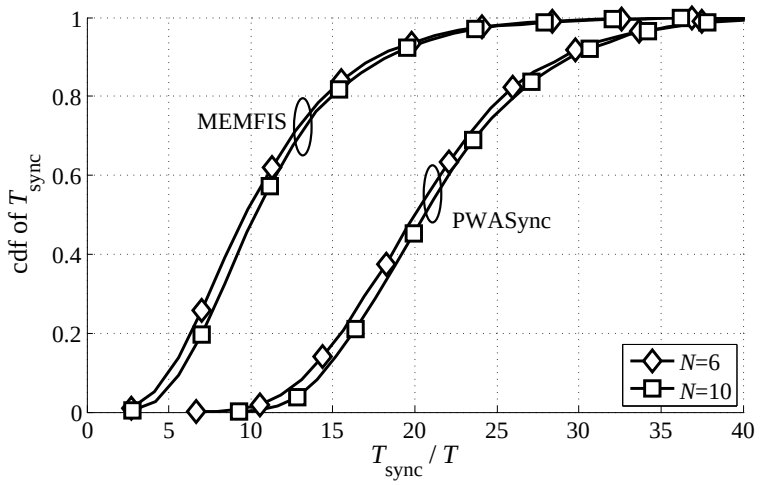
$$\Delta\tau_i[n] = \frac{\sum_{j\in\mathcal{N}_i} \Lambda^{\star}_{ij} \cdot \Delta\tau_{ij}[n]}{\sum_{j\in\mathcal{N}_i} \Lambda^{\star}_{ij}} \tag{4.39}$$

with

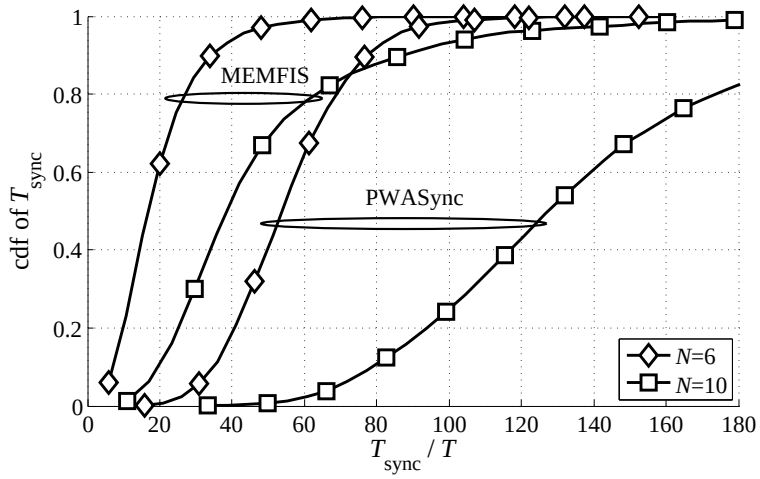
$$\Delta\tau_{ij}[n] = \widehat{\tau}_{ij}[n] - \tau_i[n] - \nu_{ij} + 2\eta \; . \tag{4.40}$$

Based on this average, the timing reference of node i is updated according to:

$$\tau_i[n+1] = \tau_i[n] + \varepsilon_{\text{PWA}} \cdot \Delta\tau_i[n] \; , \quad 0 < \varepsilon_{\text{PWA}} < 1 \; . \tag{4.41}$$



(a) Algebraic connectivity $\kappa = N$



(b) Algebraic connectivity $\kappa = 0.15$

Figure 4.18: Cdf of normalized time to synchrony T_{sync}. Comparison between MEMFIS and PWASync [SN99].

The received power level is determined by the correlator output $\Lambda_{ij}^{\star} = \Lambda_i[\theta_{ij}/T_s]$ in (4.15) at the detected reference instant $k = [\theta_{ij}/T_s]$. It was shown in [TA98] that the synchronization process is very sensitive to the initial conditions, and that propagation delays ν_{ij} need to be compensated through the adjustment factor η to avoid a constant drift.

Figure 4.18 compares PWASync and MEMFIS in two network topologies, the results being obtained with the standard parameters of Table 4.2 and $\varepsilon_{\text{PWA}} = 0.5$. In a fully-meshed network (Figure 4.18(a)), both algorithms lead to synchrony for the given parameters. MEMFIS however reaches synchronization much faster, outperforming PWASync by about 10 periods. In a sparse network (Figure 4.18(b)), MEMFIS accomplishes synchronization in average $40\,T$ and $100\,T$ faster than PWASync for a network with $N = 6$ and $N = 10$ nodes respectively. These results indicate that

MEMFIS has a better scalability with respect to the number of nodes. The reason for the superior performance of MEMFIS is intrinsically linked to the adaptation of received synchronization messages. In sparser topologies, nodes are influenced by fewer nodes, giving rise to less interactions with respect to dense topologies. In MEMFIS a single received packet can align transmitting and receiving nodes if their reference instants τ_i and τ_j are sufficiently close. On the other hand, in PWASync, reference instants are shifted by a fraction of the timing error (4.41). Therefore nodes only asymptotically converge to a common timing reference. In particular for sparse networks, as nodes are only subject to local interactions, it takes longer to reach synchronization than with MEMFIS.

4.6 Adaptation to Other Frame Structures

MEMFIS enables nodes in the network to synchronize and agree on a relative timing reference common to all by detecting the timing of transmission instants. With the transmit slot in Figure 4.3(a), this reference instant corresponds to the beginning and end of the slot. Therefore slot synchronization is achieved. The algorithm can be further extended to accommodate for different slot structures, and can also achieve frame synchronization, where a frame is a collection of slots as shown in Figure 4.19.

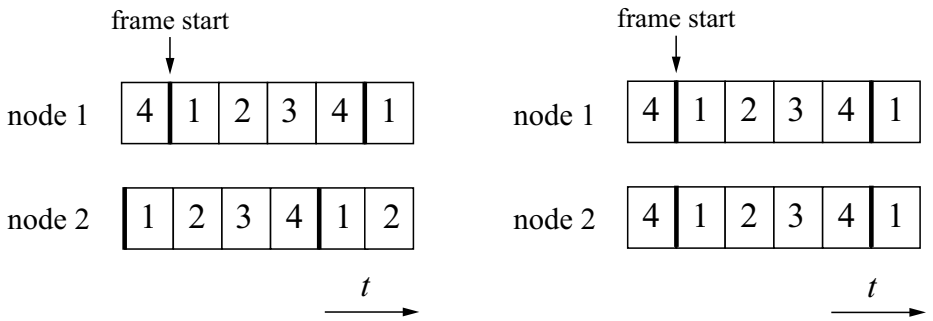


Figure 4.19: Slot and frame synchronization in TDMA systems of four slots.

In Time Division Multiple Access (TDMA), the channel allocation divides time into slots, and each node is allocated a slot where it is allowed to transmit. To this end, it is necessary for nodes to agree on the slot number further to aligning their slot start, as shown in Figure 4.19(b). To achieve this with MEMFIS, the reference instant is defined as the frame start, and nodes have to agree on a common frame start: nodes transmit the common sync-word once per frame, to indicate their frame start reference, and adjust their phase upon reception of a sync-word. The oscillation period is equal to the frame duration, e.g. four slots in Figure 4.19. This form of synchronization is further developed for the cellular network synchronization algorithm presented in Chapter 5.

The transmit slot in Figure 4.3(a) illustrates a simple structure where data and the common sync-word are multiplexed in time. In the remainder of this section, MEMFIS is adapted to fit into

IEEE 802.11 in Section 4.6.1 and into S-MAC in Section 4.6.2.

4.6.1 IEEE 802.11

The IEEE 802.11 standard is a popular standard for forming wireless local area networks. As detailed in Section 2.3.1, a synchronization protocol is put into place to perform frame synchronization, which is required for frequency hopping and the scheduling of sleep periods. Before detailing the implementation of MEMFIS to perform frame synchronization in 802.11, MEMFIS is adapted to perform slot synchronization.

Slot Synchronization In order to perform link-level synchronization, i.e. recovering the transmitter's start of transmission and frequency offset, a sync-word common to all nodes is placed at the beginning of each transmitted packet, as shown in Figure 4.20.

In OFDM mode, the sync-word is composed of two parts [IEE99]. The first part is composed of 10 identical parts, which are used for coarse time and frequency synchronization. The second part is composed of a guard interval denoted GI_2 followed by two identical parts T_1 and T_2. This preamble is followed by a data part composed of $N_{\text{data}} + 1$ OFDM symbols.

To fit MEMFIS into this slot structure, no modification of the standard is required. The common sync-word is placed as a preamble ($T_{\text{d,pre}} = 0$ in Figure 4.3(a)) and has a total duration of $T_{\text{Tx}} = 16\,\mu$s. The rest of a transmit slot is composed of control information and payload data, and has a duration of $T_{\text{data}} = (N_{\text{data}} + 1) \times 4\,\mu$s. The total duration of the transmit slot is thus equal to $T_{\text{Tx}} + T_{\text{data}}$, and is used to update the duration of a receive slot.

Frame Synchronization With some modification, MEMFIS is applied to perform frame synchronization in IEEE 802.11. The goal is for access points in the network to agree on a common frame

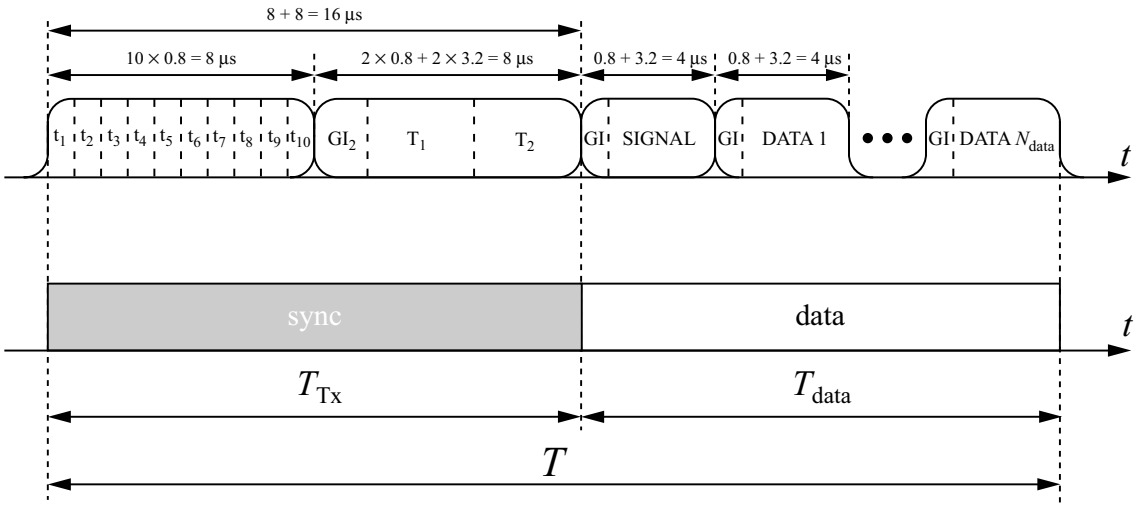
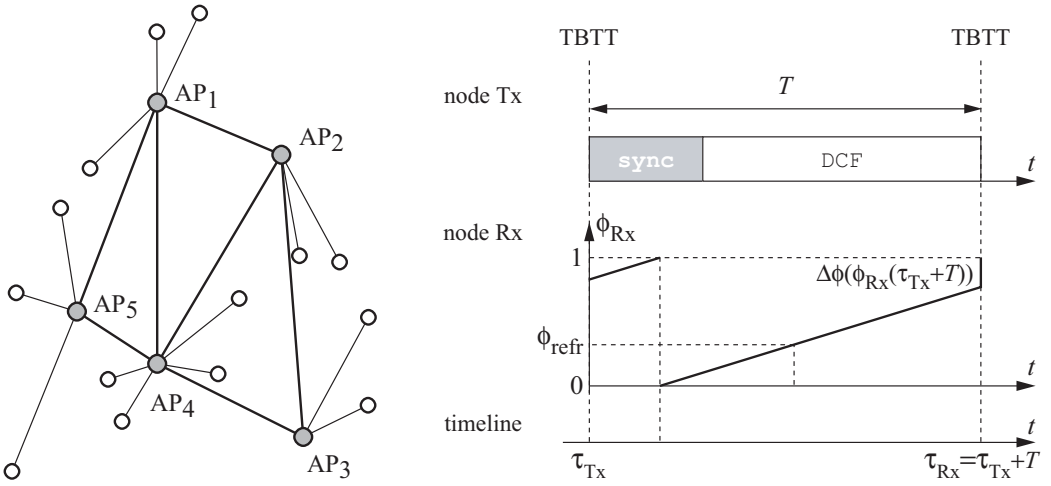


Figure 4.20: Transmit slot structure in IEEE 802.11.



(a) Synchronization among APs

(b) Synchronization of Tx and Rx

Figure 4.21: Frame synchronization in IEEE 802.11 with MEMFIS.

start (see Figure 4.21(a)), which improves coordination among the different APs, and offers simpler design for interference management and handover.

As pointed out in Section 2.3.1, the basic TSF synchronization scheme does not scale well as the network size grows, and also is not able to synchronize in dense networks. Extensions to this protocol have been presented (see Section 2.3.1), and all rely on the exchange of explicit timing information in the form of timestamps. These extensions complicate the original TSF synchronization and may add further overhead.

Applying MEMFIS to this frame structure is done by defining the reference instant as the TBTT instant, and letting nodes transmit a common sync-word when they fire, typically followed by a Distributed Coordination Function (DCF) period where nodes exchange data. This is shown in Figure 4.21(b) for transmitting node Tx. The sync-word duration in this figure is enlarged for ease of explanation.

Whilst node Tx is transmitting the sync-word, receiving node Rx fires and enters a receiving slot such that it does not transmit. This enables Rx to receive the full sync-word from Tx, and to update its phase function at the next TBTT, which results in the synchronization of both nodes. The randomness in the allocation of transmit and receive slots enables the synchronization of the nodes, and avoids node Rx to be deaf to the transmitted sync-word. Furthermore this randomness improves the detection of the sync-word as it lowers the interference level and thus increases the SINR.

4.6.2 S-MAC

An important issue in wireless sensor networks is that nodes are very energy limited. To this end, it is important to design protocols that target energy conservation and self-organization. In [YHE02], the Sensor-MAC (S-MAC) protocol was presented. It reduces energy consumption by letting nodes

periodically go to sleep. The frame structure proposed for S-MAC is composed of a listen period followed by a sleep period. In the listen period, nodes first perform synchronization in the period termed 'sync', followed by the Request To Send (RTS) period where transmitting nodes establish communication with receiving nodes. Nodes that are not communicating may go to sleep in the sleep phase, whilst communicating pairs exchange data in this period. This frame structure is presented in Figure 4.22.

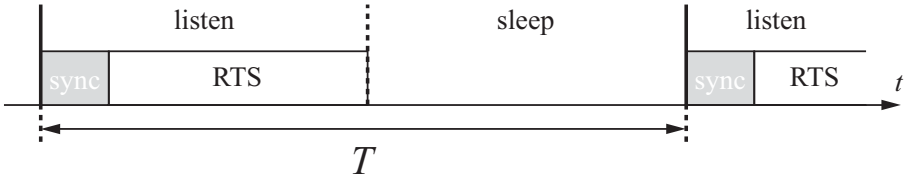


Figure 4.22: Frame structure in S-MAC.

The S-MAC protocol requires nodes to synchronize to a common frame start. This is accomplished in [YHE02] by forming virtual clusters among neighboring nodes. Nodes are synchronized within each cluster, but there is no process put into place to perform inter-cluster synchronization. Thus, MEMFIS would nicely complement S-MAC as it would perform slot synchronization seamlessly: at the beginning of each frame, some randomly selected nodes transmit a sync-word, and over time, nodes align their reference instants, thus enabling inter-cluster synchronization.

4.7 Summary

This chapter introduced MEMFIS, an adaptation of the PCO model to perform slot synchronization in decentralized wireless networks. The proposed algorithm integrates constraints from the PHY and the MAC in the following manner:

- to tackle the unavoidable delays due to the position of the sync-word in the slot, the duration of the sync-word, and the decoding time at the receiver, the proposed adaptation increases the coupling delay so that it is equal to the slot duration T when the propagation delay is neglected;
- the common sync-word is transmitted along with data. This fulfills a double requirement; as nodes cannot receive and transmit simultaneously synchronization is difficult to achieve if nodes transmit every period (see Section 4.5.4). Furthermore this enables a seamless synchronization to occur, as nodes do not distinguish between a synchronization phase and a data communication phase.

MEMFIS retains the simplicity, scalability and adaptability of the original PCO model.

Synchronization in MEMFIS relies on the detection of a common sync-word. The detection is not perfect, because the transmission of a sync-word propagates through the wireless channel, and it is also interfered by the transmission of data, of other sync-words, and by noise at the receiver. The

detection process was evaluated in terms of false alarm and detection rates and was derived for two types of detectors. Although both types are different, i.e. the cross-correlation receiver one relies on a random sequence whereas the auto-correlation receiver relies on the structure of the sync-word, they showed similar performance under some conditions. Through simulations it was shown that MEMFIS is robust to missed detections but that false alarms are more harmful because they disturb the system.

Further simulation results showed that MEMFIS presents a number of attractive features:

- it retains the simplicity, scalability and adaptability of the PCO model;
- synchronization emerges as nodes exchange data;
- the achieved throughput increases rapidly;
- only one synchronization word is needed in the network;
- explicit exchange of timestamps is avoided.
- it synchronizes faster than a competing approach based updating local timings based on the average of detected timings.

Finally extensions to fit MEMFIS into existing frame structures were presented. Distinction between slot and frame network synchronization was exemplified for the IEEE 802.11 standard, MEMFIS enabling both. The major difference resides in the definition of the reference instant; it is the beginning of a slot for slot synchronization, and the beginning of the succession of slots for frame synchronization. Several considerations need to be further taken account when sleep periods are present. If nodes are coarsely synchronized, a minimum listen time is required so that adjacent parts of the network are able to communicate and synchronize. A further adaptation is presented and examined in detail in the following chapter. Differently to MEMFIS this adaptation is targeted at cellular and base stations synchronize by hopping implicitly on the timing of the cell-edge user terminals to achieve an out-of-phase synchronization regime.

CHAPTER

5 Decentralized Inter-Base Station Synchronization

Slot synchronization is an enabling component for cellular systems. It is a prerequisite for advanced inter-cellular cooperation schemes, such as interference suppression between neighboring cells, as well as multicast and broadcasting services. The problem of inter-cell slot synchronization is to align the internal timing references of all nodes, so that base stations (BSs) and user terminals (UTs) agree on a common reference instant that marks the start of a transmission slot. In the context of cellular systems a slot is composed of a number of successive uplink and downlink frames, referred to as super-frame.

Network synchronization in cellular systems is commonly performed in a master-slave manner: BSs synchronize to an external timing reference, known as the primary reference clock, and transfer this timing to UTs. This reference clock can be acquired through the global positioning system (GPS) or through the backbone connection. The first method requires the installation of a GPS receiver at each BS, which increases costs and, more importantly, does not work in environments where GPS signals cannot be received. For high accuracy, the second method requires precise delay compensation, and the accuracy severely decreases when clocks are chained [Bre02].

Over-the-air decentralized intercell slot synchronization that avoids the need for an external timing reference was pioneered in [AAK91], and further elaborated in [LG94, IHT95]. Its basic principle is summarized as follows: a BS emits a pulse indicating its own timing reference and is receptive to pulses from surrounding BSs; internal timing references are adjusted based on the power-weighted average of received pulses (see Section 4.5.7 for more details). Conditions for convergence were derived in [TA98], which reveals that convergence and stability are tightly linked to the inter-site propagation delays between neighboring BSs. This is a critical issue, as inter-BS propagation delays are not known *a priori*. Furthermore, in [AAK91], direct communication between BSs is required, and for the exchange of synchronization pulses, a separate frequency band is assumed to be available.

In this chapter a different approach is taken based on the theory of pulse-coupled oscillators presented in Chapter 3. The proposed Cellular Firefly Synchronization (CelFSync)[1] algorithm adapts

[1]Results presented in this chapter have been performed in the framework of the IST project IST-4-027756 WINNER (World Wireless Initiative New Radio), which is partly funded by the European Union, and have been published in parts

the PCO model to account for constraints of cellular networks. CelFSync operates over-the-air, in a decentralized manner; no constraints are imposed on the availability of an external timing reference. As BSs and UTs typically transmit on successive downlink and uplink frames, two groups need to be distinguished; the BS group transmitting on the downlink and the UT group transmitting on the uplink. To facilitate the formation of two groups, two synchronization words are specified, one associated to BSs and the other to UTs. UTs transmit an uplink sync-word based on their internal timing reference, which is received by BSs to update their own timing; in return UTs adjust their timing reference upon reception of downlink sync-words from neighboring BSs. Thus, unlike [AAK91], no separate frequency band is required as sync-words are transmitted in-band with data. Moreover direct communication among BSs is not mandatory as synchronization is performed by hopping over UTs. As the downlink sync-word is mandatory for conventional cellular systems to align the timing of UTs with the BS, the only overhead for inter-BS synchronization is the insertion of the uplink sync-word. Thanks to the proposed strategy, the network is able to synchronize starting from an arbitrary misalignment, and propagation delays only affect the achieved accuracy but do not compromise the convergence to synchrony.

When considering a scenario where BSs are separated by several hundred meters up to a few kilometers, propagation delays severely affect the attainable timing accuracy. To combat this effect, CelFSync is combined with the timing advance procedure, which ensures that UT uplink transmissions arrive simultaneously at the BS. Compensating intra-cell propagation delays with the timing advance procedure, as well as selecting cell edge users to participate in CelFSync, are effective means to substantially improve the achieved inter-base station timing accuracy.

The remainder of this chapter is structured as follows. In Section 5.1 CelFSync is developed by adapting the rules that govern the synchronization of PCOs to cellular networks, and Section 5.2 combines CelFSync with timing advance to compensate the effects of propagation delays. Practical constraints regarding the implementation in cellular networks are addressed in Section 5.3, and simulation results are presented in Section 5.4 that investigate the time to convergence and the achieved accuracy for an indoor office environment as well as an urban macro-cell deployment composed of hexagonal cells.

5.1 Decentralized Inter-Cell Synchronization

This section presents an adaptation of the PCO model to perform inter-cell synchronization. To facilitate reliable exchange of reference instants in the presence of signal fading, interference and noise, long synchronization sequences that are transmitted in-band with data are considered instead of pulses. Furthermore, half-duplex transmission is considered, which implies that nodes cannot receive whilst transmitting. To this end, when two nodes transmit sync-words that partially overlap, both nodes are unable to detect the sync-word sent by the other node, referred to as *deafness* between nodes. Hence both nodes are effectively uncoupled, an effect which may severely disrupt inter-cell

in [TA08b, TA08a, TA09].

synchronization. Further accounting for constraints in cellular systems, the frame structure does not allow for overlapping downlink and uplink slots. Thus synchronized BSs and UTs should not transmit simultaneously.

The proposed Cellular Firefly Synchronization (CelFSync) scheme takes into account these fundamental constraints by resorting to an out-of-phase synchronization regime, introduced in Section 5.1.1. CelFSync relies on two synchronization sequences, one transmitted by BSs to adjust timing references of UTs, and a second one transmitted by UTs to adjust timing references of BSs, based on rules that are established in Section 5.1.2. The detection of the two distinct synchronization sequences in an asynchronous environment is discussed in Section 5.1.3. For ease of explanation, propagation delays are neglected in this section and are treated specifically in Section 5.2.

5.1.1 Synchronization Regimes

A system of PCOs is said to be synchronized when all nodes have reached a *stable state* where their internal timing references are aligned, constrained to the considered synchronization regime [Izh99]. The synchronization regime is characterized by the phase difference $\Delta = \tau_1 - \tau_2$ between two synchronized groups in the stable state, where members of the same group are perfectly aligned. Depending on the phase difference Δ, three synchronization regimes are distinguished [Izh99], as illustrated in Figure 5.1. If there is no phase shift, i.e. $\Delta = 0$, the regime is said to be *in-phase*. If the phase shift is exactly equal to half a period, i.e. $\Delta = T/2$, nodes have reached an *anti-phase* synchronization regime. Finally if the phase difference between oscillators is $\Delta \neq 0$ and $\Delta \neq T/2$ between the first and second group (and $T - \Delta$ between the second and first group), then oscillators are *out-of-phase* synchronized.

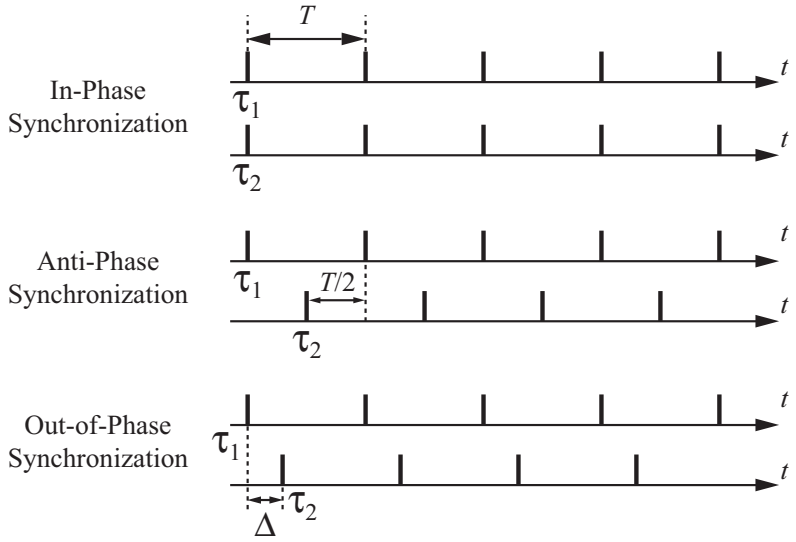


Figure 5.1: Synchronization regimes of pulse-coupled oscillators.

The in-phase regime is the most common form of synchronization; pacemaker cells pulse simultaneously to pump the heart, fireflies emit light at the same time. Anti-phase synchronization is also familiar; when walking, our legs are anti-phase synchronized: the left foot touches the ground half a period after the right one, and vice versa.

Following the frame structure of cellular systems composed of successive downlink and uplink frames, BSs are to be synchronized out-of-phase with UTs. Out-of-phase synchronization ensures that uplink and downlink transmissions in the steady state do not overlap, so that detrimental effects of deafness between nodes, inherent to half-duplex transmission, are mitigated.

5.1.2 CelFSync: Cellular Firefly Synchronization

The goal of CelFSync is to synchronize in time the transmission slots of a cellular network, so that neighboring BSs mutually align the start of the super-frame preamble. The timing information between BSs is conveyed by implicitly hopping over mobiles close to the cell edge, as exemplified in Figure 5.2. Hopping on the UT enables to extend the reception range of sync-words, and thus, allows for robust inter-cell synchronization, even when neighboring base stations do not hear one another.

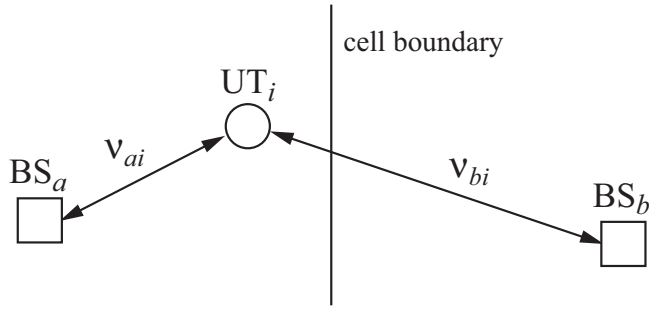


Figure 5.2: Cellular network topology with two BSs and one UT.

CelFSync adapts the PCO synchronization model so to establish an out-of-phase synchronization regime. The desired stable state is illustrated for one user terminal UT$_i$ and one base station BS$_a$ in Figure 5.3. Unlike the PCO model, instead of pulses, nodes transmit long synchronization sequences denoted by `UL_Sync` and `DL_Sync` of duration $T_{\text{UL,Sync}}$ and $T_{\text{DL,Sync}}$ respectively. For slot synchronization three states are distinguished: transmission of the sync-word, the refractory period, and the listen state. Transmission starts when a node fires (see $\tau_{\text{UT},i}$ for UT$_i$ in Figure 5.3). Half-duplex transmission is considered: when a node transmits, its receiver is switched off. After transmission of the sync-word nodes enter the refractory period, where detected sync-words are not acknowledged. In listen state nodes maintain a phase function that is adjusted upon detection of a sync-word. Key to separating nodes into two predefined groups is achieved by three types of interactions as follows.

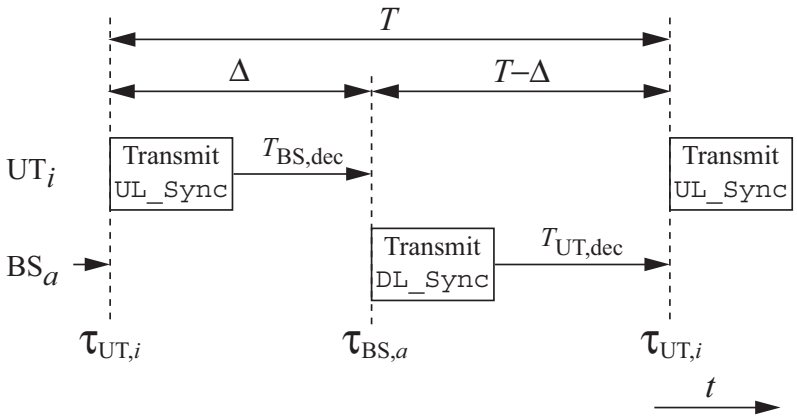


Figure 5.3: Synchronization principle of CelFSync.

UT-BS Coupling Base station BS$_a$ estimates the reference instant of UT$_i$ by detecting its sync-word UL_Sync; the estimate of this reference instant is denoted by $\widehat{\tau}_{\mathrm{UT},i}$. In order to establish the desired out-of-phase synchronization regime, BS$_a$ adjusts its phase function $\phi_{\mathrm{BS},a}$ exactly Δ seconds after UT$_i$ has fired, at instant $\theta_{\mathrm{UT},i} = \widehat{\tau}_{\mathrm{UT},i} + \Delta$. If the coupling instant $\theta_{\mathrm{UT},i}$ falls within the listen state of BS$_a$, the receiving BS increments its phase:

$$\phi_{\mathrm{BS},a}(\theta_{\mathrm{UT},i}) \rightarrow \phi_{\mathrm{BS},a}(\theta_{\mathrm{UT},i}) + \Delta\phi_{\mathrm{BS}}\left(\phi_{\mathrm{BS},a}(\theta_{\mathrm{UT},i})\right) . \tag{5.1}$$

The phase response curve $\Delta\phi_{\mathrm{BS}}$ is chosen according to (3.4), such that phase increments are strictly positive:

$$\phi + \Delta\phi_{\mathrm{BS}}(\phi) = \min\left(\alpha_{\mathrm{BS}}\,\phi + \beta_{\mathrm{BS}}, 1\right) . \tag{5.2}$$

The coupling parameters are chosen in accordance to the PCO synchronization model: $\alpha_{\mathrm{BS}} > 1$ and $0 < \beta_{\mathrm{BS}} < 1$.

The BS decoding delay $T_{\mathrm{BS,dec}}$, shown in Figure 5.3, specifies the interaction delay between the instant UL_Sync is detected at $\widehat{\tau}_{\mathrm{UT},i} + T_{\mathrm{UL,Sync}}$ and the coupling instant $\theta_{\mathrm{UT},i} = \widehat{\tau}_{\mathrm{UT},i} + \Delta$. It is an important parameter for two reasons. Firstly $T_{\mathrm{BS,dec}}$ allows for a processing delay at the receiver in order to perform link-level synchronization. Secondly $T_{\mathrm{BS,dec}}$ needs to be appropriately chosen, so that the desired out-of-phase synchronization regime is reached. As BSs fire Δ after UTs, the BS decoding delay yields:

$$T_{\mathrm{BS,dec}} = \Delta - T_{\mathrm{UL,Sync}} . \tag{5.3}$$

BS-UT Coupling The considered user terminal UT$_i$ estimates $\widehat{\tau}_{\mathrm{BS},a}$, the reference instant of BS$_a$. If the reception of DL_Sync from BS$_a$ at instant $\theta_{\mathrm{BS},a} = \widehat{\tau}_{\mathrm{BS},a} + T - \Delta$ falls within the listen state of UT$_i$, the receiving UT increments its phase:

$$\phi_{\mathrm{UT},i}(\theta_{\mathrm{BS},a}) \rightarrow \phi_{\mathrm{UT},i}(\theta_{\mathrm{BS},a}) + \Delta\phi_{\mathrm{UT}}\left(\phi_{\mathrm{UT},i}(\theta_{\mathrm{BS},a})\right) . \tag{5.4}$$

Again the phase response curve for BS-UT coupling $\Delta\phi_{\mathrm{UT}}$ is chosen according to (3.4):

$$\phi + \Delta\phi_{\mathrm{UT}}(\phi) = \min\left(\alpha_{\mathrm{UT}}\,\phi + \beta_{\mathrm{UT}}, 1\right) \tag{5.5}$$

with the coupling parameters $\alpha_{\mathrm{UT}} > 1$ and $0 < \beta_{\mathrm{UT}} < 1$. The UT decoding delay that enforces UTs to fire $T - \Delta$ after BSs is equal to (see Figure 5.3):

$$T_{\mathrm{UT,dec}} = T - \Delta - T_{\mathrm{DL,Sync}} \, . \tag{5.6}$$

Thanks to this strategy, the formation of two groups is controlled. Starting from an arbitrary initial misalignment, where all reference instants $\tau_{\mathrm{BS},a}$, $\tau_{\mathrm{UT},i}$ are randomly distributed within $[0, T]$, by following simple coupling rules, reference instants of UTs and BSs separate over time into two groups; all BSs fire Δ after UTs, and all UTs firing $T - \Delta$ after BSs. This state corresponds to the synchronized state shown in Figure 5.3. Convergence is verified through simulations in Section 5.4; by appropriately selecting the coupling parameters, it is shown that synchronization is always accomplished.

To speed up the convergence of CelFSync, two enhancements are possible, namely BS-BS and UT-UT coupling and the selection of active UTs.

BS-BS and UT-UT Coupling In case BSs can communicate directly or UTs are placed close to one another, convergence may be accelerated by allowing coupling between nodes of the same group. Moreover, the occurrence of deafness between nodes decreases, because the number of nodes that are potentially coupled is increased. As half-duplex transmission is considered, BS-BS and UT-UT couplings are useful only during the coarse synchronization phase, that is among nodes whose reference instants are misaligned by more than the sync-word length.

Phase adjustments are made similarly to (5.1) and (5.4) for BSs and UTs; however decoding delays are different, as nodes need to align in time with other nodes from their own group. Therefore the interaction delay upon detection of `DL_Sync` and `UL_Sync` needs to be equal to one period T, giving a decoding delay of $T_{\mathrm{BS\text{-}BS,dec}} = T - T_{\mathrm{DL,Sync}}$ for BSs and $T_{\mathrm{UT\text{-}UT,dec}} = T - T_{\mathrm{UL,Sync}}$ for UTs.

Active UT Selection Since uplink sync-words `UL_Sync` should be heard by multiple BSs, it is reasonable to select a subset of UTs close to the cell boundary to participate in inter-cell synchronization. Therefore, in each cell, each base station selects the N_{UT} UTs with largest propagation delays among $N_{\mathrm{UT,tot}}$ total UTs in the cell. The remaining $N_{\mathrm{UT,tot}} - N_{\mathrm{UT}}$ UTs are not active in CelFSync, and follow the timing reference dictated by their closest BS, by aligning their local clocks based on `DL_Sync`.

5.1.3 Synchronization Word Detection

CelFSync relies on the detection of transmitted `DL_Sync` and `UL_Sync` sequences. In the following, it is assumed that uplink and downlink sync-words are two different random sequences, each composed of M symbols. Sync-word detection is carried out by the link-level synchronization unit, which cross-correlates the received signal stream $y(t)$ with the known sync-word $b(t)$,

where $b(t) = b_{\mathrm{UL}}(t)$ if uplink sync-words are to be detected, and $b(t) = b_{\mathrm{DL}}(t)$ otherwise. The output of the link-level synchronization unit i is denoted by $\Lambda_i(t) = \int y(t-\tau)\, b^*(\tau)\, \mathrm{d}\tau$. The correlator output produces a series of peaks, in a similar way to the emission of pulses in the PCO model, and detection of a sync-word is declared when $\Lambda_i(t)$ exceeds the detection threshold $\Lambda^{[\mathrm{th}]}$ [SN99].

Signal fading may attenuate the received signal $y(t)$, which may result in a missed detection. The probability that reference instants $\widehat{\tau}_{\mathrm{UT},i}$ and $\widehat{\tau}_{\mathrm{BS},a}$ are correctly detected is defined as [Tre01]:

$$P_{\mathrm{d}} = \Pr\left\{ \Lambda_i(t) \geq \Lambda^{[\mathrm{th}]} \,\middle|\, \mathcal{H} \right\} \tag{5.7}$$

where $\mathcal{H}$ is the hypothesis that a sync-word is present at the receiver. On the other hand, as sync-words are transmitted in-band, cross-correlation of $s(t)$ with other sync-words, payload data or noise produces spurious peaks, so that detection of a sync-word may be declared although no sync-word is present, giving rise to a false alarm. The false alarm probability is defined as [Tre01]:

$$P_{\mathrm{fa}} = \Pr\left\{ \Lambda_i(t) \geq \Lambda^{[\mathrm{th}]} \,\middle|\, \overline{\mathcal{H}} \right\} \tag{5.8}$$

where $\overline{\mathcal{H}}$, the hypothesis that no sync-word is present at the receiver, is the complement of $\mathcal{H}$.

As described in Section 4.4, the Neyman-Pearson criterion is used to design the sync-word detector [Tre01]: the detection threshold $\Lambda^{[\mathrm{th}]}$ is set according to the desired false alarm rate P_{fa}; once $\Lambda^{[\mathrm{th}]}$ is set, the detection rate P_{d} is determined. The impact of false alarm and detection rates was studied for MEMFIS in Section 4.5.3. It was shown that false alarms have a higher impact on the convergence than missed detections $1 - P_{\mathrm{d}}$. Hence, it is necessary to maintain a sufficiently low false alarm rate.

The reliability of the link-level synchronization unit can be enhanced by increasing the length of the sync-word M. Increasing M improves the detection rate for a given false alarm rate, at the expense of higher overhead.

5.2 Compensation of Propagation Delays

The accuracy of CelFSync is limited by propagation delays, in a similar manner to the PCO model presented in Chapter 3. In an indoor environment where distances between nodes are typically small, propagation delays are negligible. However, for cellular systems where the inter-BS distance is up to a few kilometers, Section 5.2.1 reveals that propagation delays cannot be ignored. A common procedure to align uplink transmissions is the timing advance procedure, described in Section 5.2.2. Timing advance is combined with CelFSync in Section 5.2.3 to achieve a timing accuracy within a fraction of the inter-BS propagation delays.

5.2.1 Achieved Accuracy in the Stable State

After CelFSync converges and reaches a stable state, reference instants of BSs and UTs are out-of-phase synchronized (see Figure 5.3), and no phase increment occurs. In the following discussion a

sufficient refractory period (3.29) is assumed; then stability is maintained and the achieved timing accuracy in the stable state between any two nodes is bounded by (3.20).

In the presence of propagation delays, the stable state condition (3.19) in terms of the reference instants of BS_a and UT_i translates to:

$$\tau_{\mathrm{BS},a} \in \left[\tau_{\mathrm{UT},i} + \Delta - \nu_{ai}\,,\ \tau_{\mathrm{UT},i} + \Delta + \nu_{ai}\right] \tag{5.9}$$

where ν_{ai} is the propagation delay between BS_a and UT_i. When the upper bound in (5.9) is approached, i.e. $\tau_{\mathrm{BS},a} = \tau_{\mathrm{UT},i} + \Delta + \nu_{ai}$, UT_i is the forcing node that imposes its timing onto BS_a. Likewise, (5.9) approaches the lower bound, $\tau_{\mathrm{UT},i} = T - \Delta + \tau_{\mathrm{BS},a} + \nu_{ai}$, when BS_a is the forcing node that imposes its timing onto UT_i.

The effect of propagation delays on the achieved inter-BS accuracy in the stable state is analyzed with the aid of a case study, where two BSs are synchronized via one UT, as depicted in Figure 5.2. This case study resembles the discussion for a network with $N = 3$ nodes presented in Section 3.4.2. Clearly, the worst case inter-BS timing misalignment is encountered when one BS is the forcing node. Then the two end nodes BS_a and BS_b synchronize by hopping over UT_i, so that the timing misalignments over two hops add up. Applying the bound (5.9), the inter-BS accuracy is upper bounded by the sum of the BS_a to UT_i and UT_i to BS_b propagation delays:

$$|\tau_{\mathrm{BS},b} - \tau_{\mathrm{BS},a}| \leq \nu_{ai} + \nu_{bi}\,. \tag{5.10}$$

Given that in cellular networks the inter-BS distance is up to a few kilometers, propagation delays thus have a major impact on the achieved accuracy in the stable state.

5.2.2 Timing Advance Procedure

As UTs are arbitrarily distributed within the cell, the distance d_{ai} between UT_i to BS_a varies. Since propagation delays are distance dependent through $\nu_{ai} = d_{ai}/\mathrm{c}$, where c is the speed of light, the observed timing reference of BS_a measured at different UTs, denoted $\widehat{\tau}_{\mathrm{BS},a} = \tau_{\mathrm{BS},a} + \nu_{ai}$, are mutually different.

To ensure that uplink transmissions arrive simultaneously at their own base station, timing advance is a common procedure in current cellular systems [3GP99] and in wired telecommunication systems [LGHD85]. For timing advance, UT_i advances its transmission by ν_{ai}, the propagation delay to its serving BS, taken to be BS_a (see Figure 5.2). The uplink reference instant of UT_i including timing advance is given by:

$$\tau_{\mathrm{UTA},i} = \tau_{\mathrm{UT},i} - \nu_{ai}\,. \tag{5.11}$$

The propagation delay ν_{ai} may be determined by estimating the round trip delay between BS_a and UT_i [3GP08]. Upon reception of `DL_Sync` from BS_a, UT_i responds with the transmission of a random access preamble (RAP) at $\tau_{\mathrm{RAP},i} = \widehat{\tau}_{\mathrm{BS},a} + T_{\mathrm{RAP}}$. Since T_{RAP} is a constant known to BS_a, the round trip delay $2\nu_{ai}$ is determined by detecting the received timing of the RAP at BS_a. In addition, the RAP identifies UT_i, so that BS_a can distribute the estimate of ν_{ai} to UT_i.

5.2.3 CelFSync with Timing Advance

In order to combat propagation delays, CelFSync is combined with the timing advance procedure. If UT_i knows the propagation delay to its serving base station BS_a, the corresponding round trip delay of $2\nu_{ai}$ can be compensated. Owing to the multipoint-to-point topology specific to cellular networks, BS_a of cell $\mathcal{A}$ typically serves several mobiles UT_i, $i \in \mathcal{A}$, each with a specific propagation delay ν_{ai}. Hence, all timing inaccuracies, i.e. the propagation delays from BS_a to UT_i and back from UT_i to BS_a, must be compensated for at the mobile UT_i. This is accomplished by advancing both the transmitted `UL_Sync` *and* the coupling of the received `DL_Sync` at UT_i by the BS-UT propagation delay ν_{ai}.

For the following discussion, suppose that UT_i has carried out the timing advance procedure with BS_a, but its `UL_Sync` transmission is received by a neighboring base station BS_b.

UT-BS Coupling For CelFSync with timing advance, UT_i sends the uplink sync-word `UL_Sync` at the advanced reference instant $\tau_{\mathrm{UTA},i} = \tau_{\mathrm{UT},i} - \nu_{ai}$ in (5.11). Then a phase increment occurs at BS_b at instant $\theta_{\mathrm{UTA},i} = \tau_{\mathrm{UTA},i} + \Delta + \nu_{bi}$, so that (5.1) is transformed to:

$$\phi_{\mathrm{BS},b}(\theta_{\mathrm{UTA},i}) \rightarrow \phi_{\mathrm{BS},b}(\theta_{\mathrm{UTA},i}) + \Delta\phi_{\mathrm{BS}}\big(\phi_{\mathrm{BS},b}(\theta_{\mathrm{UTA},i})\big) \tag{5.12}$$

with $\theta_{\mathrm{UTA},i} = \tau_{\mathrm{UT},i} + \Delta + \nu_{bi} - \nu_{ai}$.

BS-UT Coupling For *BS-UT coupling* (5.4), the coupling is also advanced by the propagation delay. So, given that UT_i is timing aligned to BS_a but receives `DL_Sync` from BS_b, the mobile UT_i advances its coupling by ν_{ai}. Then the received `DL_Sync` from BS_b leads to a phase increment at UT_i at instant $\theta_{\mathrm{BSA},b} = \theta_{\mathrm{BS},b} - \nu_{ai}$, so that (5.4) changes to:

$$\phi_{\mathrm{UT},i}(\theta_{\mathrm{BSA},b}) \rightarrow \phi_{\mathrm{UT},i}(\theta_{\mathrm{BSA},b}) + \Delta\phi_{\mathrm{UT}}\big(\phi_{\mathrm{UT},i}(\theta_{\mathrm{BSA},b})\big) \tag{5.13}$$

with $\theta_{\mathrm{BSA},b} = \tau_{\mathrm{BS},b} + T - \nu_{ai} = \tau_{\mathrm{BS},b} - \Delta + \nu_{bi} - \nu_{ai}$.

Figure 5.4 summarizes the proposed combination of CelFSync with timing advance: UT_i starts transmission at $\tau_{\mathrm{UTA},i} = \tau_{\mathrm{UT},i} - \nu_{ai}$, so that the coupling at BS_a occurs exactly at $\tau_{\mathrm{BS},a} = \tau_{\mathrm{UT},i} + \Delta$; in return, BS_a starts transmission of its sync-word, whose decoding time is reduced at UT_i by ν_{ai} so that UT_i fires exactly $T - \Delta$ after BS_a. Hence, all entities within one cell are perfectly timing aligned, and thus, the only remaining source of timing inaccuracies is between entities of neighboring cells.

In the synchronized steady state, sync-words observed at $\tau_{\mathrm{UTA},i}$ and $\theta_{\mathrm{BSA},b}$ must fall into the refractory period, such that $\tau_{\mathrm{BS},b} \leq \tau_{\mathrm{UTA},i} < \tau_{\mathrm{BS},b} + T_{\mathrm{refr}}$ for UT-BS coupling, and $\tau_{\mathrm{UT},i} \leq \theta_{\mathrm{BSA},b} < \tau_{\mathrm{UT},i} + T_{\mathrm{refr}}$ for BS-UT coupling.

The steady state accuracy between BS_b and UT_i is bounded by the two extreme cases when either BS_b or UT_i takes the role of the forcing node. In case UT_i is forcing, the observed timing at BS_b yields $\tau_{\mathrm{BS},b} = \tau_{\mathrm{UT},i} + \Delta + \nu_{bi} - \nu_{ai}$. Otherwise, if BS_b is forcing, the timing imposed on UT_i amounts to $\tau_{\mathrm{UT},i} = \tau_{\mathrm{BS},b} - \Delta + \nu_{bi} - \nu_{ai}$. This means that the achieved accuracy in the steady state

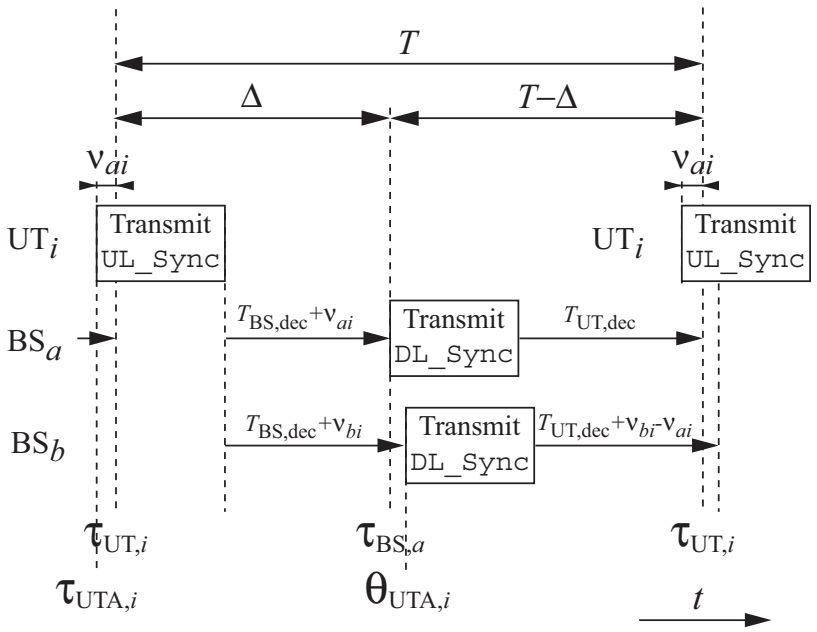


Figure 5.4: Combination of CelFSync with timing advance.

between BS_b and UT_i is bounded by:

$$\tau_{BS,b} \in \left[\tau_{UT,i} + \Delta - |\nu_{bi} - \nu_{ai}|\,,\ \tau_{UT,i} + \Delta + |\nu_{bi} - \nu_{ai}|\right]. \tag{5.14}$$

Therefore combining timing advance with CelFSync always achieves an accuracy that is bounded by the *difference* of UT-BS propagation delays.

In order to analyze the achieved inter-BS accuracy, the case study depicted in Figure 5.2 and discussed in Section 5.2.1 is revisited. Given that UT_i is time aligned to BS_a, that is $\tau_{UTA,i} = \tau_{UT,i} - \nu_{ai}$, the only remaining source of inaccuracies is the link from UT_i to BS_b, so that the UT-BS accuracy bound (5.14) can be directly applied. Substituting $\tau_{UT,i} = \tau_{BS,a} - \Delta$ into (5.14), the inter-BS accuracy between BS_a and BS_b over two hops is bounded to:

$$|\tau_{BS,b} - \tau_{BS,a}| \leq |\nu_{ai} - \nu_{bi}|\,. \tag{5.15}$$

Provided that UT_i is located near the cell boundary, its propagation delays to BS_a and BS_b are similar, so that the difference $|\nu_{ai} - \nu_{bi}|$ is much smaller than the individual delays ν_{ai} and ν_{bi}. This is in sharp contrast to the achieved accuracy without timing advance in (5.10), which is bounded by the sum of propagation delays. Increasing the UT density per cell $N_{UT,tot}$ increases the probability of selected UTs to be close to the cell edge, which has the appealing effect that the inter-BS accuracy (5.15) improves. The accuracy bound is extended to multiple UTs in the following section.

The working principle of CelFSync including timing advance is summarized as follows

- UT_i connects to the BS with the strongest received signal strength, assumed to be BS_a.

- UT_i aligns its timing to BS_a by carrying out a timing advance procedure, as described in Section 5.2.2.

- If identified as active, UT_i emits `UL_Sync` at reference instants $\tau_{\mathrm{UTA},i}$ in (5.11), and adjusts its phase $\phi_{\mathrm{UT},i}$ upon reception of `DL_Sync` according to (5.13).

5.2.4 Achieved Accuracy for Multiple UTs

In the following the inter-BS accuracy bound (5.15) is extended to multiple UTs. Active UTs that are timing aligned to BS_a and BS_b are associated to cells $\mathcal{A}$ and $\mathcal{B}$, respectively. Entities within cells $\mathcal{A}$ and $\mathcal{B}$ are perfectly timing aligned, such that $\tau_{\mathrm{BS},a} = \tau_{\mathrm{UT},i} + \Delta$, $\forall\, i \in \mathcal{A}$, and $\tau_{\mathrm{BS},b} = \tau_{\mathrm{UT},i} + \Delta$, $\forall\, i \in \mathcal{B}$. In line with the discussion in Section 5.2.3, timing misalignments between entities belonging to different cells are bounded by four extreme cases: either UTs in cell $\mathcal{A}$ or $\mathcal{B}$ are forcing by imposing their timing reference $\tau_{\mathrm{UT},i}$ to neighboring BS; alternatively either BS_a or BS_b forces UTs in neighboring cells.

If UTs in cell $\mathcal{A}$ are forcing, then UT_i, $i \in \mathcal{A}$ with the earliest timing reference $\tau_{\mathrm{UT},i}$ imposes its time reference to BS_b, such that $\tau_{\mathrm{BS},b} = \min_{i\in\mathcal{A}} \left\{ \tau_{\mathrm{UT},i} + \Delta + \nu_{bi} - \nu_{ai} \right\}$. Since $\tau_{\mathrm{BS},a} = \tau_{\mathrm{UT},i} + \Delta$ is valid for all entities within cell $\mathcal{A}$, the timing reference of BS_b yields:

$$\tau_{\mathrm{BS},b} = \tau_{\mathrm{BS},a} + \min_{i\in\mathcal{A}} \left\{ \nu_{bi} - \nu_{ai} \right\}. \tag{5.16}$$

Now consider the case when BS_b forces UTs in cell $\mathcal{A}$. For BS-UT coupling (5.13) the reference instant of BS_b causes a phase adjustment at UT_i at instant $\theta_{\mathrm{BSA},b} = \tau_{\mathrm{BS},b} - \Delta + \nu_{bi} - \nu_{ai}$. Since $\tau_{\mathrm{BS},a} = \tau_{\mathrm{UT},i} + \Delta$ generally holds for all entities in cell $\mathcal{A}$, the UT_i, $i \in \mathcal{A}$, whose UT-BS propagation delays minimize the difference $\nu_{bi} - \nu_{ai}$ receives the earliest $\theta_{\mathrm{BSA},b}$. This UT then triggers BS_a and in turn the remaining UTs of cell $\mathcal{A}$, and thus determines the accuracy between BS_b and the UTs in cell $\mathcal{A}$. When BS_b is forcing UTs in cell $\mathcal{A}$, the timing reference of BS_a therefore yields:

$$\tau_{\mathrm{BS},a} = \tau_{\mathrm{BS},b} + \min_{i\in\mathcal{A}} \left\{ \nu_{bi} - \nu_{ai} \right\}. \tag{5.17}$$

Due to symmetry, the remaining two cases, i.e. when either UTs of cell $\mathcal{B}$ force BS_a or BS_a forces UTs of cell $\mathcal{B}$, are obtained by exchanging a with b, and $\mathcal{A}$ with $\mathcal{B}$ in (5.16) and (5.17). This yields the inter-BS accuracy bound for CelFSync with timing advance between two cells:

$$\left| \tau_{\mathrm{BS},b} - \tau_{\mathrm{BS},a} \right| \leq \max \left\{ \left| \min_{i\in\mathcal{B}} \{ \nu_{ai} - \nu_{bi} \} \right|, \left| \min_{i\in\mathcal{A}} \{ \nu_{bi} - \nu_{ai} \} \right| \right\}. \tag{5.18}$$

If UTs are timing aligned to the BSs with the shortest distance, the difference $\nu_{bi} - \nu_{ai}$, for $i \in \mathcal{A}$ and $\nu_{ai} - \nu_{bi}$, for $i \in \mathcal{B}$, will always be positive. Hence, the bound (5.18) improves with growing numbers of UTs per cell $|\mathcal{A}|$ and $|\mathcal{B}|$. Asymptotically, when $|\mathcal{A}|, |\mathcal{B}| \to \infty$, the accuracy approaches zero, so that the effect of propagation delays is perfectly compensated. This trend is confirmed by the simulation results presented in Section 5.4.2, which show that the achieved inter-BS accuracy significantly improves as the number of users per cell $N_{\mathrm{UT,tot}}$ increases.

5.3 Implementation Aspects

In order to integrate CelFSync into a cellular mobile radio standard, several practical constraints need to be taken into consideration. Constraints regarding the frame structure and the chosen duplexing scheme are addressed in this section.

5.3.1 Frame Structure

CelFSync is implemented and verified based on the frame structure taken from the specifications of the WINNER[2] system concept [ADvH+07]. Consecutive downlink and uplink slots constitute one frame, and a number of successive frames form one super-frame of duration T. One uplink and one downlink sync-word `UL_Sync` and `DL_Sync` are placed into the super-frame with a relative spacing of Δ, as illustrated in Figure 5.1.

The downlink sync-word `DL_Sync` allows UTs to synchronize to its BS and is therefore essential in cellular networks. Unlike `DL_Sync`, the insertion of the uplink sync-word `UL_Sync` adds overhead, as `UL_Sync` is typically not required in current cellular networks. Fortunately this overhead is modest as `UL_Sync` is typically transmitted with low rate. For instance, for the WINNER system, the respective durations for super-frame and `UL_Sync` are 5.8 ms and 45 μs. Hence the resulting overhead is less than 1% [ADvH+07].

5.3.2 Acquisition and Tracking Modes

An intrinsic property of PCO synchronization is that coupling between nodes effectively shortens period T. However, cellular systems typically rely on a fixed frame structure, which specifies the way uplink and downlink slots are arranged to exchange payload data. To this end, whilst the reception of payload data is still ongoing, CelFSync may shorten the period of two successive reference instants to $T' \leq T$, which effectively shortens the duration of the super-frame.

As long as the effective period T' is only slightly shortened, such that $T - T' \leq \varepsilon_{\text{at}}$, insertion of a guard time with duration $T_{\text{G}} > \varepsilon_{\text{at}}$ ensures that reception of payload data is completed before a sync-word is transmitted. The condition $T - T' \leq \varepsilon_{\text{at}}$ corresponds to the *tracking mode* in the steady synchronization state, where small offsets due to clock skews, leading to deviations of the natural oscillation period T between nodes, are compensated.

In case of coarse timing misalignments between cells, i.e. $T - T' > \varepsilon_{\text{at}}$, the network is in *acquisition mode*. Potential conflicts in acquisition mode are avoided by:

(i.) suspending payload data transmission while inter-cell synchronization is in progress;

(ii.) shortening the super-frame duration to $T_{\text{sf}} < T$.

Scheme (i.) does not allow for exchange of payload data before CelFSync has reached a steady state. Given that a steady state is likely to be maintained for hours or even days, while CelFSync typically converges within a fraction of a second or so, the loss in system throughput due to suspended

[2]Wireless World Initiative New Radio, URL: www.ist-winner.org

data transmissions may be acceptable. For instance, scheme (i.) is applied to facilitate the synchronization procedure in the wireless LAN standard IEEE 802.11 [IEE99, ZL04]: periodically, data transfer is preempted, and the access point transfer its TSF clock value to the network participants (see Section 2.3.1).

Scheme (ii.) avoids conflicts by forcing the effective period T' to be at least as long as T_{sf}. By doing so, continuous exchange of payload data is maintained, at the expense of reducing the throughput during acquisition by about $(T - T_{\mathrm{sf}})/T$.

5.3.3 Duplexing Scheme

CelFSync is applicable to both time division duplex (TDD) and frequency division duplex (FDD) - based systems. Nodes adjust their internal clocks based on received sync-words; whether the uplink and downlink sync-words are transmitted on different frequency bands or not is irrelevant. The discussion in this chapter targets half-duplex transmission, where nodes cannot receive and transmit at the same time, applicable to TDD and half-duplex FDD. Full-duplex FDD benefits CelFSync, since nodes can transmit and receive simultaneously, which eliminates deafness due to missed sync-words whilst transmitting.

5.3.4 Imposing an Absolute Timing Reference

An inherent problem of any distributed synchronization procedure is that nodes agree on a *relative time reference* that is valid only among the considered nodes and has no external tie. Such a relative reference is opposed to an *absolute time reference* such as the Coordinated Universal Time, which is provided by GPS for example (see Section 2.2). Furthermore, as the size of the network increases, it becomes increasingly difficult to synchronize the entire network in a completely decentralized manner. To avoid this difficulty, in [TA07] a scenario was considered where only a few nodes have access to an absolute time reference. The PCO model was extended such that these master nodes impose an absolute time reference to the entire network, even though the number of master nodes was only a small fraction of the total number of nodes in the network. Furthermore, the behavior of *normal nodes* that do not have access to an absolute time reference, is not modified at all.

Applied to CelFSync, a subset of BSs have access to an absolute time reference. These master BSs emit downlink sync-words `DL_Sync` with a slightly shortened period $T_{\mathrm{ma}} < T$, and are *not* receptive to sync-words from other nodes [TA07]. Neighboring cells then align their reference instants following the synchronization rules outlined in Section 5.1.2. It was demonstrated in [TA07] that for $0.9\,T \leq T_{\mathrm{ma}} < T$, arbitrarily large networks are reliably synchronized. By doing so the problem of synchronizing large networks with a distributed algorithm is reduced to synchronizing a number of cells (typically up to 2 or 3 tiers) around a master BS.

5.4 Performance Evaluation

To evaluate the performance of CelFSync two deployment scenarios are considered: first an indoor office scenario in Section 5.4.1; and second a macro cell deployment modeled by an hexagonal cell structure in Section 5.4.2 [KMH+07]. All nodes transmit with the same power P_s. The propagation channel between nodes i and j is modeled as a distance dependent path loss channel. Node j receives the transmission of a node i at a distance d_{ij} with power $P_s d_{ij}^{-\chi}$, where χ is the path loss exponent. The signal-to-noise-plus-interference ratio (SINR) of a received sync-word is composed of the received power of the sync-word, divided by the level of interference plus thermal noise with power σ^2. The detection threshold is set for a given false alarm rate, which enables the computation of the detection probability P_d for each received sync-word as a function of the current SINR (see Section 5.1.3). Unless otherwise stated, the parameters shown in Table 5.1 are used in the simulations.

Both environments impose different strains on CelFSync. In the indoor environment, sync-words are subject to a high level of interference from other transmitting UTs. In the outdoor environment, the large distance between UTs and BSs result in higher channel attenuations, creating a more sparsely connected network, which implies that network synchronization is to be carried out over multiple hops.

In both scenarios, Monte-Carlo simulations are conducted for $5,000$ sets of initial conditions: all BSs initially commence with uniformly distributed internal timing references, while UTs are locally synchronized to their closest BS. Synchronization is declared when two groups have formed, so that

Parameter	Symbol	Default value	
		indoor	macro-cell
Transmit power	P_s	10 dBm	
Path-loss exponent	χ	4	3
Noise level	σ^2	-93 dBm	
False alarm rate	P_{fa}	10^{-4}	
Sync-word length	M	32 symbols	
Durations of sync-words	$T_{UL,Sync}$, $T_{DL,Sync}$	45 μs	
Super-frame duration	T	5.89 ms	
Out-of-phase offset	Δ	0.11 ms	
BS refractory	$T_{BS,refr}$	2.33 ms	
BS coupling	α_{BS}	1.15	
	β_{BS}	0.01	
UT refractory	$T_{UT,refr}$	2.33 ms	
UT coupling	α_{UT}	1.3	
	β_{UT}	0.01	
Number of BSs	N_{BS}	4 BSs	19 BSs
Number of active UTs	N_{UT}	15 UTs	3 UTs/cell

Table 5.1: Default simulation parameters.

reference instants of UTs are aligned and out-of-phase synchronized with reference instants of BSs, with a relative timing difference of Δ.

5.4.1 Indoor Office Environment

An indoor office with two corridors and ten offices on each side is considered. This setting was defined for the local area scenario in WINNER [AAC+06]. The network topology with $N_{\mathrm{BS}} = 4$ BSs and $N_{\mathrm{UT}} = 15$ UTs participating in CelFSync is depicted in Figure 5.5. The selected UTs (marked as bold circles), can communicate directly with all BSs (marked as squares). UTs that do not participate in the network synchronization procedure do not transmit `UL_Sync`, and adjust their slot

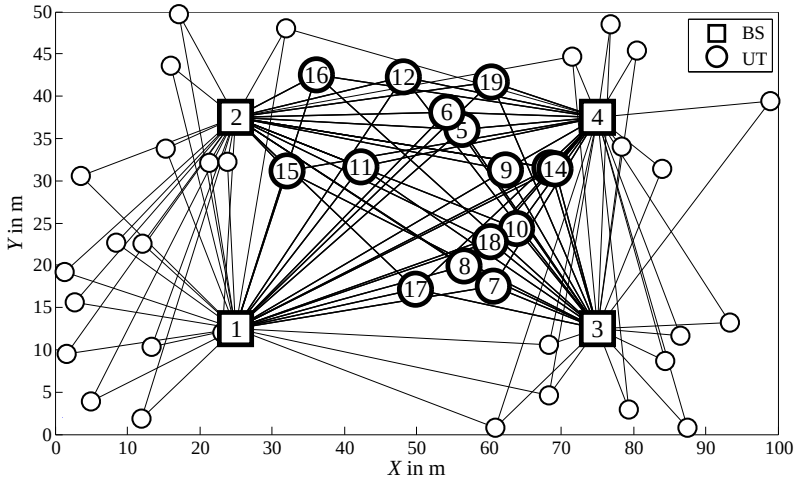


Figure 5.5: Considered indoor network topology.

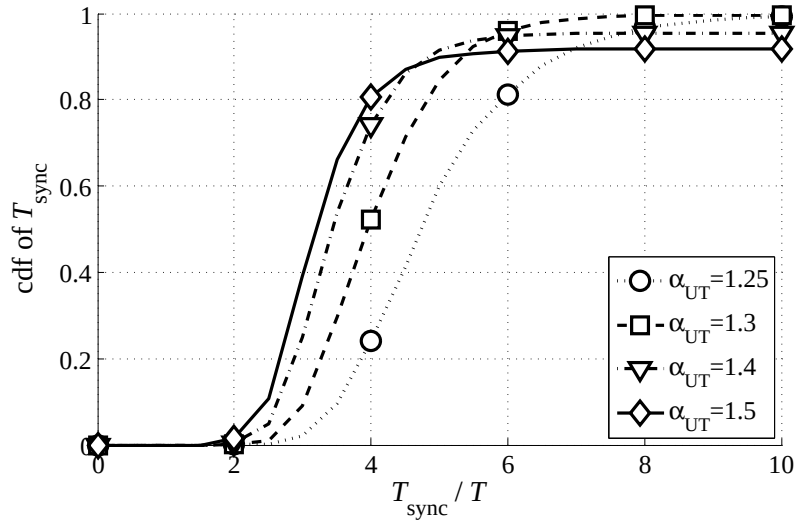


Figure 5.6: Cdf of the normalized time to synchrony in the considered indoor environment when varying the BS-UT coupling.

oscillator based on received `DL_Sync`.

Results plotted in Figure 5.6 elaborate on the time taken for the entire network to synchronize. The time to synchrony T_{sync} is normalized to the duration of a super-frame T. Figure 5.6 plots the cdf of the normalized time to synchrony for different values of the BS-UT coupling factor α_{UT}.

The performance of the proposed inter-BS synchronization scheme can be controlled by the coupling factor α_{UT}. For a high coupling value, i.e. $\alpha_{\text{UT}} > 1.3$, synchronization is reached quickly, but convergence to a synchronized stable state is not always achieved. The fraction of initial conditions that do not converge to this state is due to deafness among nodes: some part of the network transmits partially overlapping `DL_Sync` and `UL_Sync` sequences, and due to the half-duplex assumption, some nodes are thus not able to synchronize. The deafness probability increases with the coupling factor α_{UT}, and for $\alpha_{\text{UT}} = 1.5$, it is approximately 10%. If the coupling is low, i.e. $\alpha_{\text{UT}} \leq 1.3$, synchronization is always reached within $T_{\text{sync}} = 10$ periods, and for $\alpha_{\text{UT}} = 1.3$, 80% of initial conditions lead to synchrony within $T_{\text{sync}} = 5$ periods. This is encouraging given the fact that deafness among nodes does not occur when $\alpha_{\text{UT}} \leq 1.3$, even though nodes start from a random initial timing reference. Setting α_{UT} sufficiently low reduces the absorption limit (3.6), which allows nodes to receive more sync-words in the synchronization phase. This lowers the deafness probability, and enables the network to synchronize starting from any initial timing misalignment.

Interestingly the time to synchrony is much lower with CelFSync than with MEMFIS. The comparison is not completely fair, because the connectivity and network diameter of the local area network in Figure 5.5 are respectively higher and lower than the default values considered in Section 4.5. A strong argument that motivates the quicker convergence of CelFSync is that in CelFSync, base stations and active user terminals transmit their sync-word at the beginning of *every* period. This naturally augments the number of interactions and thus speeds up convergence. The effect of deafness is minimal compared to MEMFIS thanks to the out-of-phase synchronization regime that is achieved. With MEMFIS, such a behavior is not possible, because nodes synchronize in-phase and thus synchronized nodes that transmit simultaneously do not hear each other, unlike synchronized nodes in CelFSync.

5.4.2 Macro Cell Deployment

For the macro cell deployment, an hexagonal cell structure is considered as shown in Figure 5.7. One or two tiers of BSs are placed around a center BS, resulting in a network of $N_{\text{BS}} = 7$ and $N_{\text{BS}} = 19$ BSs, respectively, each of radius of $d_{\text{cell}} = 1\,\text{km}$. The number of active UTs per cell, N_{UT}, specifies the number of UTs that participate in CelFSync. Among the $N_{\text{UT,tot}}$ UTs randomly placed in each cell, the N_{UT} UTs closest to the cell edge are selected as active.

Time To Synchrony

In a similar manner to Figure 5.6, results plotted in Figure 5.8 depict the time to synchrony of CelFSync in an hexagonal cell deployment for $N_{\text{BS}} = 7$ BSs and $N_{\text{BS}} = 19$ BSs. Coupling among UTs is also considered with strength $\alpha_{\text{UT-UT}} = 1.05$.

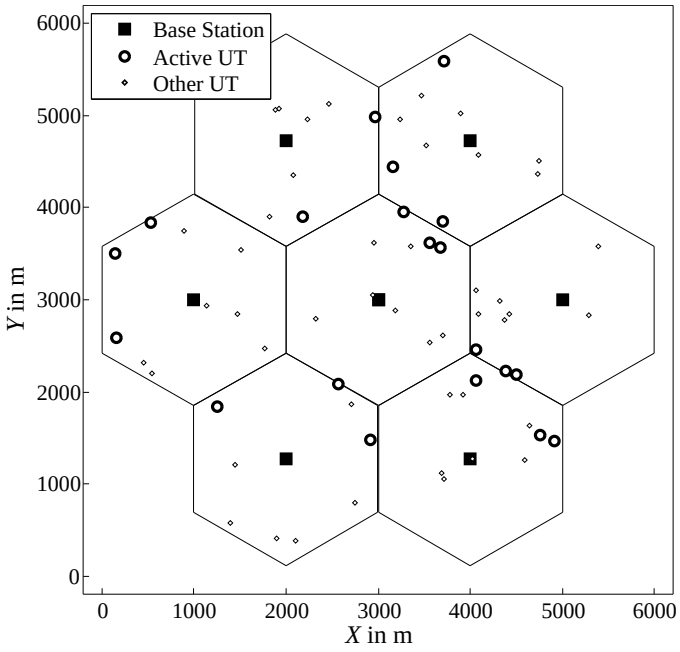


Figure 5.7: Macro cell network topology composed of $N_{\mathrm{BS}} = 7$ hexagonal cells with $N_{\mathrm{UT}} = 3$ active UTs per cell.

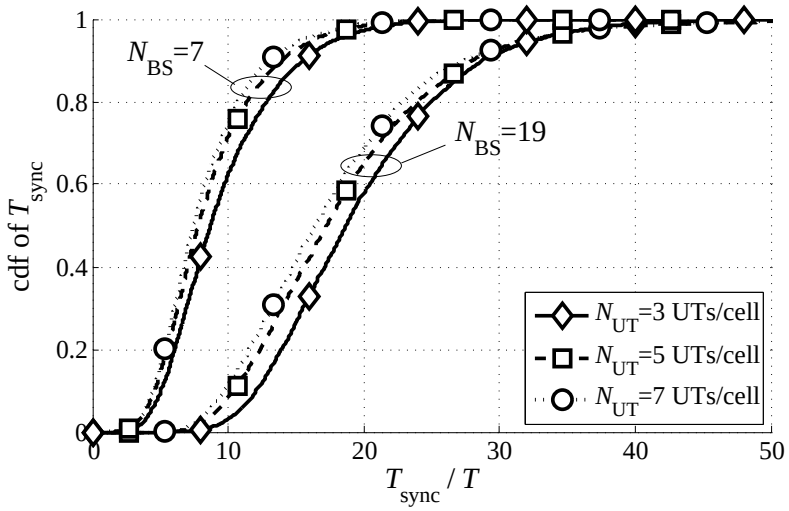


Figure 5.8: Cdf of the normalized time to synchrony for an hexagonal cell deployment scenario with $N_{\mathrm{BS}} = 7$ and $N_{\mathrm{BS}} = 19$ base stations.

As expected, networks of $N_{\mathrm{BS}} = 19$ BSs converge less rapidly than smaller networks of $N_{\mathrm{BS}} = 7$ BSs. This degradation is due to the increase in network diameter from 4 hops to 8 hops. Moreover, the number of UTs per cell participating to CelFSync, N_{UT}, does not significantly change the time to synchrony, and a synchrony rate of 80% is achieved within $12T$ when $N_{\mathrm{BS}} = 7$ BSs and within $25T$ when $N_{\mathrm{BS}} = 19$ BSs. In all cases, a synchronization rate of 100% is achieved

within $T_{\text{sync}} = 50$ periods, which implies that deafness between nodes, due to partially overlapping sync-words, does not corrupt the convergence of CelFSync.

Achieved Inter-BS Accuracy

While in an indoor environment propagation delays are typically negligible, the opposite is true for the macro-cell deployment (5.10). The achieved inter-BS accuracy $\varepsilon_{ab} = |\tau_{\text{BS},b} - \tau_{\text{BS},a}|$ of CelFSync including timing advance is verified in Figure 5.9 for various node densities $N_{\text{UT,tot}}$. Simulations are conducted over 100 random network topologies, each with 200 sets of initial conditions. It is assumed that UTs are timing aligned with their closest BS, and that the number of active UTs per cell is set to $N_{\text{UT}} = 3$ UTs per cell.

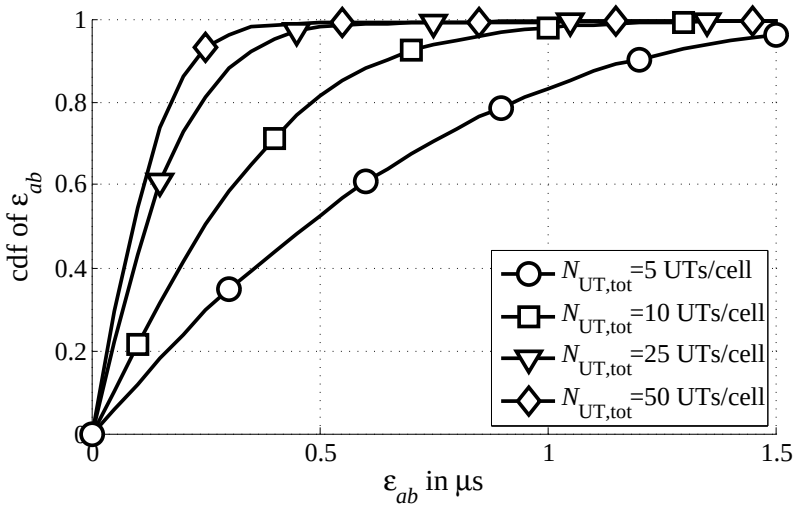


Figure 5.9: Achieved inter-BS accuracy for $N_{\text{BS}} = 19$ BSs with timing advance and $N_{\text{UT}} = 3$ active UTs.

As the accuracy bound (5.15) suggests, the inter-BS accuracy ε_{ab} is significantly improved as the node density $N_{\text{UT,tot}}$ increases. Augmenting $N_{\text{UT,tot}}$ increases the probability for selected UTs to be close to the cell edge, which decreases the delay difference $\nu_{bi} - \nu_{ai}$ in (5.15). For a UT density equal or higher than $N_{\text{UT,tot}} \geq 25$ UTs per cell, the achieved accuracy is bounded by $\varepsilon_{ab} < 0.5\,\mu$s. This is a significant achievement as the propagation delay for an inter-BS distance of $2\,d_{\text{cell}} = 2\,$km is $\nu_{ab} \approx 6.67\,\mu$s.

5.5 Summary

This chapter studied the application of the PCO model to cellular networks. The original algorithm was modified so that base stations and user terminals achieve an out-of-phase synchronization regime: the DL sync-word of BSs causes synchronized UTs to transmit their UL sync-word Δ after, and vice versa, this UL sync-word transmission causes surrounding base stations to fire $T - \Delta$

later. Thanks to this regime nodes synchronize significantly faster than with MEMFIS, because they transmit every period. With the proposed CelFSync algorithm, a local area wireless network composed of 4 base stations and 15 user terminals is always able to synchronize within 10 periods.

In large-scale networks where propagation delays are typically non-negligible, the timing advance procedure, common in current cellular networks, was combined with CelFSync to combat the effect of large propagation delays. By compensating intra-cell propagation delays with timing advance together with selecting cell edge users to participate in CelFSync, the detrimental effects of large propagation delays are substantially reduced. Simulation results demonstrated that the achieved inter-BS timing accuracy is always below $1\,\mu$s when at least 10 users are randomly distributed per cell, which corresponds to approximately 15% of the direct propagation delay for an inter-BS spacing of 2 km.

CHAPTER

6 Conclusions and Open Issues

Natural synchronization phenomena are fascinating and were the inspiration for the synchronization studies presented in this thesis. The PCO model, which describes phenomena where oscillators interact through discrete interactions, served as a basis for the studies on self-organized slot synchronization. Its properties are very attractive and were exploited through two adaptations, termed MEMFIS and CelFSync, the former targeting completely decentralized wireless networks and the latter targeting cellular networks.

Summary

Chapter 2 defined the slot synchronization problem with regards to synchronization in communication networks. Synchronization is a requirement at many levels, and for slot synchronization, clocks placed at different locations need to agree on a common slot start by overcoming their initial misalignment.

Chapter 3 presented the PCO model. The robustness and characteristics of this model were verified under imperfect conditions such as coupling delay, clock drift, and a meshed network topology. The introduction of coupling delays reduces the achieved accuracy, but the accuracy among neighboring nodes is always equal or below the coupling delay, and in over 90% of cases, non-neighboring nodes are synchronized within twice the direct coupling delay. Studies of the PCO model were completed and facilitated by the introduction of a metric that quantifies the local synchronization state. This metric was applied to the studies of two imperfections. The first study looked at the PCO model in the presence of frequency drift and showed that even in sparsely connected meshed networks, nodes synchronize as long as a maximum drift is observed. The second study extended to meshed networks a stability condition derived for two nodes.

Chapter 4 presented MEMFIS, an adaptation of the PCO model to perform slot synchronization in decentralized wireless networks. Applying the PCO model to wireless networks, constraints on the PHY layer, i.e. delays and half-duplex transmission, and on the MAC layer, i.e. transmission and reception of packets, need to be taken into account. MEMFIS tackles the PHY constraints by

utilizing a long synchronization word instead of a pulse as the coupling means, and by delaying to the coupling so that interactions occur with a delay equal to the slot duration T when neglecting propagation delays. Constraints on the MAC are acknowledged by multiplexing in time the common sync-word with data, so that a sync-word is transmitted only when a packet is scheduled for transmission. The convergence analysis showed that MEMFIS has similar dynamics as the PCO model and converges in meshed networks if the graph representing the network remains connected for a sufficient time. The performance evaluation showed that MEMFIS behaves similarly to the PCO model in terms of scalability, that the algorithm is robust with regards to missed detections, and that it outperforms a competing approach based on updating local timings according to the average of detected timings.

Chapter 5 studied the application of the PCO model to cellular systems. The original algorithm was modified to attain an out-of-phase synchronization regime where base stations transmit simultaneously on downlink frames, and the transmission of user terminals is shifted in time so that they transmit simultaneously on uplink frames. With the proposed decentralized CelFSync algorithm, a local area wireless network composed of 4 base stations and 15 user terminals is always able to synchronize within 10 periods. In large-scale networks where propagation delays are typically non-negligible, the timing advance procedure, common in current cellular networks, was combined with CelFSync to combat the effect of propagation delays. By compensating intra-cell propagation delays with timing advance together with selecting cell edge users to participate in CelFSync, the detrimental effects of large propagation delays are substantially reduced. Simulation results demonstrated that the achieved inter-BS timing accuracy is always below $1\,\mu$s when at least 10 users are randomly distributed per cell, which corresponds to approximately 15% of the direct propagation delay for an inter-BS spacing of 2 km.

Open Issues

The present thesis will hopefully serve as a basis for future studies on self-organized synchronization. In the following several possible extensions are presented. A distinction is made between direct extensions, which would further deepen the knowledge into the synchronization process, and indirect extensions, which would extend this work to research topics diverging from self-organized synchronization in wireless networks.

Direct Extensions Several issues regarding MEMFIS and CelFSync are interesting for future studies.

Hardware implementation of the synchronization algorithms is a natural next step. The algorithm anticipates several issues that occur in the implementation, namely the presence of delays and integrating the sync-word into a given frame structure, but implementation requires to take into account further constraints such as fitting the sync-word detector into a protocol stack and composing with hardware requirements. Several hurdles that need to be considered during the implementation

include:

- a major requirement of the proposed algorithms is the detection of the timings of transmitted sync-words. This requires sync-words to be detected within the decoding delay, so that the corresponding coupling instant is effective within the following period;
- accessing the received signal to perform the detection of sync-words may not be evident, and it is more likely that the signal at the detector has already been sampled and demodulated. In this case there is an inherent delay between the receiver input and the sync-word detector, and this delay needs to be measured and possibly calibrated, so that it can be incorporated into the decoding delay.

An interesting issue concerns the mathematical treatment of self-organized synchronization. Although the formal treatment of self-organized systems is not straightforward, e.g. it took many years from the two node proof of Peskin [Pes75] to the many node proof of Mirollo and Strogatz [MS90], mathematical studies of these systems are very useful. They bring a better understanding of the system dynamics and are needed to understand the emergence of pattern formations and to investigate the robustness of algorithms with regards to, for example, coupling parameters, the impact of delays, or dynamic network topologies. In the literature, the Kuramoto model is more studied than the PCO model, e.g. [YS99, MB02, ES03], because its interactions are continuous, which makes the model more tractable. Nevertheless some interesting work attempts to treat delays in the PCO model analytically, e.g. [EPG95b, TWG02], but do not integrate a refractory period.

Another open issue includes the robustness against malicious attacks that would attempt to disrupt the synchronization process. This study requires to find the optimal disruption strategy, i.e. a malicious node attempts to prevent nodes from synchronizing by transmitting sync-words in a way that normal nodes cannot agree on a common reference instant. Furthermore this study could examine the detection of a malicious or malfunctioning node by normal nodes, and finding means to improve the robustness against attacks.

Finally, a possibly very interesting area for future research on self-organized synchronization is its combination with complementary areas of research. Some areas benefiting from self-organized synchronization include:

- the PCO model could serve as a primitive for self-organized scheduling, optimization, or computation in wireless networks. Some interesting applications in this direction have been proposed to perform low-complexity scheduling [PHS09] and to compute averages [DL09];
- synchronization is very important for positioning. The position accuracy being directly proportional to the synchronization accuracy, a very tight synchronization bound is required [LAK99]. This is commonly obtained through GPS, which provides positioning information by estimating the delays at the receiver with at least four satellites. Indoor positioning would greatly benefit from self-organized synchronization, as it cannot rely on GPS and would make use of local timing information to evaluate the delays.

Indirect Issues This thesis focused on applications of the PCO synchronization model to wireless networks, accounting and compensating for delays so that nodes always synchronize and agree on a common timing reference. This study could potentially be applied and benefit other domains. To summarize and generalize, the MEMFIS and CelFSync principles are applicable to a system of nodes presenting a periodic behavior and coupled through discrete and delayed events. Below are some examples of possible deviations.

The developed adaptations could be used in circuit systems. In these systems, components typically rely on a central clock, and conduct operations based on the edge of the clock. Distributing several clocks in the system provides robustness against failure, but requires clocks to synchronize to a common slot start or edge transition.

Moving to vehicular technologies, the organization of traffic lights presents similarities to the problems encountered when applying the PCO model to wireless networks. Each light can be seen as an oscillator going through three states, i.e. green, yellow, and red, and the aim is to coordinate lights so that the overall length of traffic jams is reduced. If the traffic lights are not directly connected through some network, implicit signaling among them is provided by the cars moving from one light to the other. As cars do not move at great speed, coupling among lights is delayed, and nodes are coupled only if cars are present. These conditions are similar to MEMFIS, which could provide a basis for future work. The application in this case presents several interesting challenges, because lights may have very different oscillation periods, and the goal is significantly different from the in-phase synchronization regime achieved by MEMFIS as it requires lights to form a pattern so that the total length of traffic jams is minimized.

List of Own Publications

Journal Papers

[1] A. Tyrrell, G. Auer, and C. Bettstetter. Emergent Slot Synchronization in Wireless Networks. *Accepted for publication in IEEE Transactions on Mobile Computing*, June 2009.

[2] A. Tyrrell and G. Auer. Biologically Inspired Intercellular Slot Synchronization. *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, Article ID 854087, 12 pages, January 2009.

[3] A. Tyrrell and G. Auer. Decentralized Slot Synchronization for Cellular Mobile Radio. *DoCoMo Technical Journal*, 10:60-67, 2008.

Book Chapter

[1] A. Tyrrell, G. Auer, and C. Bettstetter. *Advances in Biologically Inspired Information Systems*, chapter Biologically Inspired Synchronization for Wireless Networks, pages 47-62. Ed. F. Dressler and I. Carreras. Springer, 2007.

Patents

[1] PA151-EP1936837, Self-Organized Synchronization for Cellular Systems, December 2006.

[2] PA123-EP1852998, Meshed Emergent Firefly Synchronization, February 2006.

[3] PA109-EP1802013, Imposing a Reference onto Self-Organized Synchronization, November 2005.

[4] PA095-EP1744472, Delay-Tolerant Firefly Synchronization, May 2005.

Conference Papers

[1] A. Tyrrell, G. Auer, and C. Bettstetter. A Synchronization Metric for Meshed Networks of Pulse-Coupled Oscillators. In *Proceedings of the 3rd International Conference on Bio-Inspired Models of Network, Information, and Computing Systems (BIONETICS 2008)*, pages 1-7, Hyogo, Japan, November 2008.

[2] A. Tyrrell, G. Auer, and C. Bettstetter. On the Accuracy of Firefly Synchronization with Delays. In *Proceedings of the 1st International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2008)*, pages 1-5, Aalborg, Denmark, October 2008. Awarded the Best Student Paper Prize of the conference.

[3] A. Tyrrell and G. Auer. Decentralized Inter-Base Station Synchronization Inspired from Nature. In *Proceedings of the 68th IEEE Vehicular Technology Conference (VTC 2008-Fall)*, pages 1-5, Calgary, Canada, September 2008.

[4] L. Sanguinetti, A. Tyrrell, M. Morelli, and G. Auer. On the Performance of Biologically-Inspired Slot Synchronization in Multicarrier Ad Hoc Networks. In *Proceedings of the 67th IEEE Vehicular Technology Conference (VTC 2008-Spring)*, pages 21-25, Marina Bay, Singapore, May 2008.

[5] A. Tyrrell and G. Auer. Imposing a Reference Timing onto Firefly Synchronization in Wireless Networks. In *Proceedings of the 65th IEEE Vehicular Technology Conference (VTC 2007-Spring)*, pages 222-226, Dublin, Ireland, April 2007.

[6] A. Tyrrell, G. Auer, and C. Bettstetter, Fireflies as Role Models for Synchronization in Ad Hoc Networks. In *Proceedings of the 1st International Conference on Bio-Inspired Models of Network, Information and Computing Systems (BIONETICS 2006)*, pages 1-7, Cavalese, Italy, December 2006.

[7] A. Tyrrell, G. Auer, and C. Bettstetter. Synchronization Inspired from Nature for Wireless Meshed Networks. In *Proceedings of the 2nd International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2006)*, pages 1-4, Wuhan, China, September 2006.

[8] A. Tyrrell, G. Auer, and C. Bettstetter. Firefly Synchronization in Ad Hoc Networks. In *Proceedings of the 4th MiNEMA workshop*, pages 1-4, Leuven, Belgium, February 2006.

List of Acronyms and Symbols

Abbreviations

3GPP	3rd Generation Partnership Project
AP	Access Point
ARQ	Automatic Repeat Request
ASP	Automatic Self-time-correcting Procedure
ATSF	Adaptive Timer Synchronization Function
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BS	Base Station
cdf	cumulative distribution function
CelFSync	Cellular Firefly Synchronization
CGPM	General Conference on Weights and Measures
CP	Cyclic Prefix
Cs	Caesium
DCF	Distributed Coordination Function
DNA	Deoxyribonucleic Acid
e.g.	(*Lat.*) exempli gratia (*Eng.* for example)
FTSP	Flooding Time Synchronization Protocol
GI	Guard Interval
GMT	Greenwich Mean Time

GPS	Global Positioning System
GSM	Global System for Mobile Communications
i.e.	(*Lat.*) id est (*Eng.* that is)
IBSS	Independent Basic Service Set
IMAP	Internet Message Access Protocol
IP	Internet Protocol
LAN	Local Area Network
LLC	Logical Link Control
MAC	Medium Access Control
MEMFIS	Mobile Emergent Firefly Synchronization
ML	Maximum Likelihood
MTSP	Multihop Time Synchronization Protocol
NTP	Network Time Protocol
OFDM	Orthogonal Frequency Division Multiplexing
OSI Model	Open Systems Interconnection Reference Model
PCO	Pulse-Coupled Oscillator
pdf	Probability Density Function
PHY	Physical Layer
PLL	Phase-Locked Loop
PRC	Phase Response Curve
PWASync	Power-Weighted Average Synchronization
RAP	Random Access Preamble
RBS	Reference Broadcast Synchronization
RNC	Radio Network Controller
ROC	Receiver Operating Characteristic
RTS	Request To Send

S-ALOHA	Slotted-ALOHA
S-MAC	Sensor-MAC
s.t.	such that
S/H	Sample and Hold
SDH	Synchronous Digital Hierarchy
SFO	Sampling Clock Frequency Offset
SI	International System of Units
SINR	Signal-to-Interference-plus-Noise Ratio
SONET	Synchronous Optical Network
STO	Symbol Timing Offset
TAI	International Atomic Time
TBTT	Target Beacon Transmission Time
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
TDP	Time-Diffusion Synchronization Protocol
TPSN	Timing-Sync Protocol for Sensor Networks
TSF	Timer Synchronization Function
TSS	Time-Stamp Synchronization
UMTS	Universal Mobile Telecommunications System
UT	User Terminal
UTC	Coordinated Universal Time
UWB	Ultra Wide Band
WSN	Wireless Sensor Network

Symbols

a_i	Data symbol transmitted by node i
$a_{j,p}$	Data symbols in the pth OFDM symbol of transmitting node j

$\widehat{a}_i$	Estimated signal transmitted by node i
$\mathbf{A}(\mathcal{G})$	Adjacency matrix of graph $\mathcal{G}$
$\mathcal{A}$	Set of nodes in the cell of BS_a
A_{ij}	Amplitude of the received signal of node j at node i
a_{ij}	Element of the adjacency matrix $\mathbf{A}(\mathcal{G})$
b_q	Flipping value for the qth segment of the OFDM synchronization word
b	Synchronization word
b_{DL}	Signal of sync-word `DL_Sync`
b_{UL}	Signal of sync-word `UL_Sync`
$\mathbf{C}$	Vector of internal clocks c_i
c	Speed of light
c_i	Clock of oscillator i
d_{cell}	Cell radius
d_{ij}	Distance between nodes i and j
d_{r}	Maximum distance between two connected nodes
$\mathbf{D}(\mathcal{G})$	Degree matrix of graph $\mathcal{G}$
$d_{\max}$	Area size
D_{pre}	Number of symbols in the data preamble
`DL_Sync`	Downlink synchronization word
E_i	Energy component of the synchronization metric of [MBL03]
$\mathcal{E}$	Edge set of the considered graph $\mathcal{G}$
f_{d}	Frequency drift
f_{n}	Phase noise
f_0	Nominal frequency
F	Curve describing the fixed points in the dynamics of two nodes
$\mathcal{G}$	Graph describing the considered network

g	Shaping filter impulse response
$\mathbf{h}_{ij}$	Channel impulse response between transmitting node j and receiving node i
h_{ij}	Tap of the channel impulse response between transmitting node j and receiving node i
h_{f}	Firing map
$\overline{\mathcal{H}}$	Hypothesis that no sync-word was transmitted
$\mathcal{H}$	Hypothesis that a sync-word was transmitted
h_{R}	Return map
I_{ϕ}	Interval of initial conditions not leading to immediate absorptions
$\mathcal{J}_i$	Set of transmitting nodes whose transmissions are observed by node i
K	Number of symbols per slot
L	Length of the channel impulse response
$\mathbf{L}(\mathcal{G})$	Laplacian matrix of graph $\mathcal{G}$
M	Number of symbols of the synchronization word
$\mathcal{N}_i$	Set of neighbors of node i
N	Number of nodes in the system
N_{B}	Number of data block per OFDM frame
N_{s}	Number of subcarriers per OFDM frame
N_{UT}	Number of user terminals per cell participating in CelFSync
$N_{\mathrm{UT,tot}}$	Total number of user terminals per cell
n_i	Slot number at node i
P	Number of samples in each segment of the OFDM synchronization word
P_{d}	Detection probability
P_{fa}	False alarm probability
$P_{\mathrm{fa}}^{[\mathrm{slot}]}$	False alarm probability per slot
P_i	Average received power at node i

P_s	Transmit power
Q	Number of flipped segment composing the OFDM synchronization word
$\mathbf{R}$	Vector of local synchronization metrics r_i
$\mathcal{R}_i$	Set of nodes whose sync-word transmission creates a peak in slot n_i at node i
r	Kuramoto synchronization index
r_i	Local synchronization metric of node i
$\mathbf{s}$	Elementary segment of the OFDM synchronization word
s_i	State variable at node i switching between reception and transmission
$\mathcal{S}$	Set describing a system of N oscillators
$\mathcal{S}_\ell$	Set of oscillators absorbed by the reception of a pulse
$\mathbf{s}_q$	qth segment of the OFDM synchronization word
t_i	Time process of oscillator i
T	Natural period of oscillator and duration of a slot
T_B	OFDM bandwidth
$T_\mathrm{BS,dec}$	Decoding delay for base stations
T_dec	Decoding delay
$T_\mathrm{DL,Sync}$	Duration of `DL_Sync`
$T_\mathrm{d,pre}$	Duration of the data postamble
$T_\mathrm{d,pre}$	Duration of the data preamble
T'	Effective period duration during synchronization
T_G	Guard interval duration
T_m	Matched filter integration duration
T_RAP	Duration of a random access preamble
T_s	Sampling period
T_sf	Reduced super-frame duration during acquisition
T_sync	Time to synchrony

T_{Tx}	Duration of the synchronization word
$T_{\text{UL,Sync}}$	Duration of `UL_Sync`
$T_{\text{UT,dec}}$	Decoding delay for user terminals
t	Time variable
$T_{\max}$	Maximum period duration in the presence of frequency drift
$T_{\min}$	Minimum period duration in the presence of frequency drift
`UL_Sync`	Uplink synchronization word
$\mathcal{V}$	Vertex set of the considered graph $\mathcal{G}$
w_i	Noise at receiver i
$\widehat{x}_i$	Estimated symbols at receiving node i
x_i	Signal transmitted by node i
$x_{j,p}$	pth OFDM symbol of transmitting node j
y_i	Signal received by node i
y_{ij}	Component from node j in the received signal of node i
z_i	Noise at the correlator of node i without a sync-word
$\bar{z}_i$	Noise at the correlator of node i in the presence of a sync-word

Greek letters

α	Slope of the phase response curve
α_{BS}	Slope of $\phi_{\text{BS},a}$
α_{UT}	Slope of $\Delta\phi_{\text{UT}}$
$\alpha_{\text{UT-UT}}$	Slope of the PRC for UT-UT coupling
β	Value of the phase response curve at $\phi_i = 0$
β_{BS}	Initial value of $\phi_{\text{BS},a} + \Delta\phi_{\text{BS}}$
β_{UT}	Initial value of $\phi_{\text{UT},i} + \Delta\phi_{\text{UT}}$
γ_{ij}	SINR of the transmission of node j at node i
γ_{th}	SINR threshold for data decoding

Δ	Out-of-phase shift
$\Delta\phi$	Phase response curve
$\Delta\phi_{\mathrm{BS}}$	Phase response curve at base stations
$\Delta\phi_{\mathrm{UT}}$	Phase response curve at user terminals
$\varepsilon_{\mathrm{at}}$	Minimum period reduction for deciding between an acquisition or a tracking mode
ε	Coupling strength in the Kuramoto model
ζ_i	Frequency drift of oscillator i
$\zeta_{\max}$	Maximum frequency drift
$\theta_{\mathrm{BSA},b}$	Coupling instant at receiving user terminal UT_i with timing advance
$\theta_{\mathrm{UTA},i}$	Coupling instant at receiving base station BS_b with timing advance
$\theta_{\mathrm{BS},a}$	Coupling instant at UT_i upon reception of `DL_Sync` from BS_a
$\theta_{\mathrm{UT},i}$	Coupling instant at BS_a upon reception of `UL_Sync` from UT_i
θ_{ij}	Detection instant of a peak at the sync-word detector of node i from the transmission of node j
ϑ_{ij}	Frequency offset between transmitting node j and receiving node i
κ	Algebraic connectivity
Λ_i	Correlator output of the sync-word detector of node i
$\Lambda_{ij}^{\star}$	Detected instant for the peak caused by the transmission of node j to node i
$\Lambda_i^{[\mathrm{th}]}$	Comparison threshold of the detector of node i
λ	Normalized detection threshold
λ_{pkt}	Mean packet arrival rate
μ_i	Amplitude component of the synchronization metric of [MBL03]
$\nu_{\max}$	Maximum propagation delay between connected nodes
ν_{ij}	Propagation delay between nodes i and j
σ^2	Variance of the noise at the receiver

σ_z^2	Variance of the correlator output when no sync-word is present
$\bar{\sigma}_z^2$	Variance of the correlator output in the presence of a sync-word
$\widehat{\tau}_{ij}$	Estimate of τ_j at node i
$\tilde{\tau}_{ij}$	Beginning of reception at node i of the signal of node j
τ_i	Reference instant of oscillator i
$\tau_{\mathrm{UTA},i}$	Transmission start of UT i with timing advance
$\tau_{\mathrm{RAP},i}$	Transmission start of a RAP by UT_i
$\tau_{\mathrm{UT},i}$	Reference instant of UT i
$\widehat{\tau}_{\mathrm{BS},a}$	Estimated firing instant of BS_a at receiving base station UT_i
$\widehat{\tau}_{\mathrm{UT},i}$	Estimated firing instant of UT_i at receiving base station BS_a
Φ_i	Total phase of oscillator i
$\phi_{\mathrm{BS},a}$	Phase of base station BS_a
ϕ_{refr}	Refractory phase after firing
$\phi_{\mathrm{UT},i}$	Phase of user terminal UT_i
ϕ	Mean phase of the considered oscillators
ϕ_i	Phase function of oscillator i
ϕ_ℓ	Phase absorption limit
ϕ_{fix}	Unique fixed point in a system of two pulse-coupled oscillators
χ	Pathloss exponent
ψ_{ij}	Timing offset between transmitting node j and receiving node i

Bibliography

[3GP99] 3GPP TS 05.10. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Radio Subsystem Synchronization. Technical report, December 1999.

[3GP04a] 3GPP TS 25.402. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Synchronization in UTRAN Stage 2. Technical report, December 2004.

[3GP04b] 3GPP TS 25.411. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; UTRAN Iu interface layer 1. Technical report, December 2004.

[3GP04c] 3GPP TS 25.431. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; UTRAN Iub interface Layer 1. Technical report, December 2004.

[3GP08] 3GPP TS 36.211. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation. Technical report, May 2008.

[AAC+06] M. Abaii, G. Auer, Y. Cho, I. Cosovic, M. Döttling, K. George, L. Hamm, P. Jesus, S. Kyriazakos, V. Jungnickel, A. Mihovska, M. Olsson, A. Osseiran, K. Pantelis, D. Schulz, C. Silva, M. Sternad, T. Svensson, Y. Wan, C. Wijting, and Y. Zhu. D6.13.7: WINNER II Test Scenarios and Calibration Cases Issue 2, December 2006.

[AAK91] Y. Akaiwa, H. Andoh, and T. Kohama. Autonomous decentralized inter-base-station synchronization for TDMA microcellular systems. In *Proceedings of the 41st IEEE Vehicular Technology Conference (VTC 1991)*, pages 257–262, St. Louis, MO, USA, May 1991.

[Abr77] N. Abramson. The throughput of packet broadcasting channels. *IEEE Transactions on Communications*, 25(1):117–128, January 1977.

[ADvH+07] G. Auer, K. Doppler, J. von Häfen, N. Johansson, K. Kalliojärvi, T. Lestable, J. Luo, J. Nyström, M. Olsson, A. Osserian, M. Sternad, T. Svensson, and C. Wijting. D6.13.14: WINNER II System Concept Description, December 2007.

[Ari99] Aristotle. *Physics*, chapter VI:9, page 239b5. Oxford University Press, 1999.

[Aug38] Augustine. *The Confessions of St. Augustine*, chapter XI, page 235. J.H. Parker, 1838.

[AV30] A.A. Andronov and A.A. Vitt. On mathematical theory of entrainment (in Russian). *Zhurnal Prikladnoi Fiziki (Journal on Applied Physics)*, 6(4):3–17, 1930.

[BBCH81] J. Buck, E. Buck, J.F. Case, and F.E. Hanson. Control of flashing in fireflies. V. Pacemaker synchronization in *pteroptyx cribellata*. *Journal of Comparative Physiology A*, 144(3):630–633, September 1981.

[BBJ07] O. Babaoglu, T. Binci, and M. Jelasity. Firefly-inspired heartbeat synchronization in overlay networks. In *Proceedings of the 1st IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007)*, pages 77–86, Boston, MA, USA, July 2007.

[BC00] P.C. Bressloff and S. Coombes. Dynamics of strongly coupled spiking neurons. *Neural Computation*, 12(1):91–129, January 2000.

[BCdS97] P.C. Bressloff, S. Coombes, and B. de Souza. Dynamics of a ring of pulse-coupled oscillators: Group-theoretic approach. *Physical Review Letters*, 79(15):2791–2794, October 1997.

[Big94] N.L. Biggs. *Algebraic Graph Theory*. Cambridge University Press, 1994.

[Bla15] K.G. Blair. Luminous insects. *Nature*, 96(Dec. 9):411–415, December 1915.

[Bla73] B.E. Blair. Time and frequency dissemination: An overview of principles and techniques. Technical report, Time and Frequency Division. Institute for Basic Standards. National Bureau of Standards, 1973.

[Bre02] S. Bregni. *Synchronization of Digital Telecommunications Networks*. Wiley, 1st edition, 2002.

[Buc88] J. Buck. Synchronous rhythmic flashing of fireflies. II. *The Quarterly Review of Biology*, 63(3):265–289, September 1988.

[Bur] Bureau International des Poids et Mesures. BIPM - International Atomic Time, http://www.bipm.org/en/scientific/tai/tai.html, accessed on 20 march 2009.

[Bur69] Bureau International des Poids et Mesures. Comptes Rendus de la 13e CGPM (1967/68). Technical report, 1969.

[Bus05] S. Bush. Low-energy sensor network time synchronization as an emergent property. In *Proceedings of the 14th International Conference on Computer Communications and Networks (ICCCN 2005)*, pages 93–98, San Diego, CA, USA, October 2005.

[CCS06] W. Chu, C.J. Colbourn, and V.R. Syrotiuk. The effects of synchronization on topology-transparent scheduling. *Wireless Networks*, 12(6):681–690, November 2006.

[CDF+01] S. Camazine, J.-L. Deneubourg, N.R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems*. Princeton, 1st edition, 2001.

[CGL00] A. Chandra, V. Gummalla, and J.O. Limb. Wireless medium access control protocols. *IEEE Communication Surveys and Tutorials*, 3:2–15, September 2000.

[CPB+06] E. Carlson, C. Prehofer, C. Bettstetter, H. Karl, and A. Wolisz. A distributed end-to-end reservation protocol for IEEE 802.11-based wireless mesh networks. *IEEE Journal on Selected Areas in Communications*, 24(11):2018–2027, November 2006.

[CRW+73] W. Crowther, R. Rettberg, D. Walden, S. Ornstein, and F. Heart. A system for broadcast communications: Reservation-ALOHA. In *Proceedings of the Sixth Hawaii International Conference on System Sciences*, pages 371–374, Hawaii, USA, January 1973.

[CWH07] G.-N. Chen, C.-Y. Wang, and R.-H. Hwang. Multi-hop time synchronization protocol for IEEE 802.11 wireless ad hoc networks. *Journal of Information Science and Engineering*, 23(4):969–983, July 2007.

[DBR06] L. Dai, P. Basu, and J. Redi. An energy efficient and accurate slot synchronization scheme for wireless sensor networks. In *Proceedings of the 3rd International Conference on on Broadband Communications, Networks, and Systems (BROADNETS 2006)*, pages 1–8, San José, CA, USA, October 2006.

[DDP03] A. Daliot, D. Dolev, and H. Parnas. Self-stabilizing pulse synchronization inspired by biological pacemaker networks. In *Proceedings of the Symposium on Self-Stabilizing Systems (SSS 2003)*, pages 32–48, San Francisco, CA, USA, June 2003.

[DEA06] I. Demirkol, C. Ersoy, and F. Alagoz. MAC protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, 44(4):115–121, April 2006.

[DL09] K. Deng and Z. Liu. Distributed computation of averages over wireless sensor networks through synchronization of data-encoded pulse-coupled oscillators. *International Journal of Wireless Information Networks*, 16(1-2):51–58, March 2009.

[EGE02] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI 2002)*, pages 147–163, Boston, MA, USA, December 2002.

[EPG95a] U. Ernst, K. Pawelzik, and T. Geisel. Synchronization induced by temporal delays in pulse-coupled oscillators. *Physical Review Letters*, 74(9):1570–1573, February 1995.

[EPG95b] U. Ernst, K. Pawelzik, and T. Geisel. Synchronization induced by temporal delays in pulse-coupled oscillators. *Phys. Rev. Lett.*, 74(9):1570–1573, February 1995.

[ERLH02] A. Ebner, H. Rohling, M. Lott, and R. Halfmann. Decentralized slot synchronization in highly dynamic ad hoc networks. In *Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications (WPMC 2002)*, pages 494–498, Honolulu, HI, USA, October 2002.

[ES03] M.G. Earl and S.H. Strogatz. Synchronization in oscillator networks with delayed coupling: A stability criterion. *Physical Review E*, 67:036204, March 2003.

[GG98] F. Gini and G.B. Giannakis. Frequency offset and symbol timing recovery in flat-fading channels: A cyclostationary approach. *IEEE Transactions on Communications*, 46(3):400–411, March 1998.

[GKS03] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, pages 138–149, Los Angeles, CA, USA, November 2003.

[HH95] A.V.M. Herz and J.J. Hopfield. Earthquake cycles and neural reverberations: Collective oscillations in systems with pulse-coupled threshold elements. *Physical Review Letters*, 75(6):1222–1225, August 1995.

[HS03] Y.-W. Hong and A. Scaglione. Time synchronization and reach-back communications with pulse-coupled oscillators for UWB wireless ad hoc networks. In *Proceedings of the IEEE Conference on Ultra Wideband Systems and Technologies (UWBST 2003)*, pages 190–194, Reston, Virginia, USA, November 2003.

[HS05] Y.-W. Hong and A. Scaglione. A scalable synchronization protocol for large scale sensor networks and its applications. *IEEE Journal on Selected Areas in Communications*, 23(5):1085–1099, May 2005.

[HS06] A. Hu and S.D. Servetto. On the scalability of cooperative time synchronization in pulse-connected networks. *IEEE Transactions on Information Theory*, 52(6):2725–2748, June 2006.

[IEE99] IEEE Standard for Information Technology - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[IEE03] IEEE Standard for Information Technology - Local and Metropolitan Area Networks - Specific Requirements Part 15.4, 2003.

[IEE05] IEEE Standard for Information Technology - Local and Metropolitan Area Networks - Specific Requirements Part 15.1, 2005.

[IHT95] S. Izumi, A. Hirukawa, and H. Takanashi. PHS inter-base-station frame synchronization technique using UW with experimental results. In *Proceedings of the 6th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 1995)*, pages 1128–1132, Toronto, Canada, September 1995.

[IMA03] RFC 3501: specification of IMAP version 4 revision 1, 2003.

[Inc] CollabNet Inc. Subversion, http://subversion.tigris.org, accessed on 18 december 2008.

[Izh98] E.M. Izhikevich. Phase models with explicit time delays. *Physical Review E*, 58(1):905–908, July 1998.

[Izh99] E.M. Izhikevich. Weakly pulse-coupled oscillators, FM interactions, synchronization, and oscillatory associative memory. *IEEE Transactions on Neural Networks*, 10(3):508–526, May 1999.

[JFR99] J. Jespersen and J. Fitz-Randolph. *From Sundials to Atomic Clocks*. U.S. Department of Commerce Technology Administration, 1999.

[Joh01] S. Johnson. *Emergence. The connected lives of ants, brains, cities, and software*. Scribner, 2001.

[Joh04] S. Johannessen. Time synchronization in a local area network. *IEEE Control Systems Magazine*, 24(2):61–69, April 2004.

[JSL02] Y.K. Jeong, O.-S. Shin, and K.B. Lee. Fast slot synchronization for intercell asynchronous DS/CDMA systems. *IEEE Transactions on Wireless Communications*, 1(2):353–360, April 2002.

[Kle96] W.J. Klepczynski. *Global Positioning System: Theory and Applications, Vol.2*, chapter 17: GPS for Precise Time and Time Interval Measurement, pages 483–500. American Institute of Aeronautics and Astronautics, 1996.

[KMH+07] P. Kyösti, J. Meinilä, L. Hentilä, X. Zhao, T. Jämsä, C. Schneider, M. Narandziæ, M. Milojeviæ, A. Hong, J. Ylitalo, V.-M. Holappa, M. Alatossava, R. Bultitude, Y. de Jong, and T. Rautiainen. D1.1.2: WINNER II Channel Models, December 2007.

[KS97] V. Kirk and E. Stone. Effect of a refractory period on the entrainment of pulse-coupled integrate-and-fire oscillators. *Physics Letters A*, 232(1-2):70–76, February 1997.

[Kur84] Y. Kuramoto. *Chemical Oscillations, Waves, and Turbulence*. Springer-Verlag, 1984.

[KVM+07] K. Kusume, R. Vilzmann, A. Müller, C. Hartmann, and G. Bauch. A multiuser detection perspective on medium access control in ad hoc networks. In *Proceedings of the IEEE Global Telecommunications Conference (Globecom 2007)*, pages 801–806, Washington, DC, USA, November 2007.

[KWM04] S. Kashihara, N. Wakamiya, and M. Murata. Implementation and evaluation of scalable and robust scheme for data gathering in wireless sensor networks. In *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, pages 279–280, Baltimore, MD, USA, November 2004.

[KXC+00] Z. Kostic, Q. Xiaoxin, K. Chawla, F.C. Li, and J. Chuang N. Sollenberger. Dynamic frequency hopping in cellular systems with network assisted resource allocation. In *Proceedings of the 51th IEEE Vehicular Technology Conference (VTC 2000-Spring)*, pages 2459–2463, Tokyo, Japan, May 2000.

[LAK99] W. Lewandowski, J. Azoubib, and W.J. Klepczynski. GPS: primary tool for time transfer. *Proceedings of the IEEE*, 87(1):163–172, January 1999.

[Lam78] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, July 1978.

[Lau17] P. Laurent. The supposed synchronal flashing of fireflies. *Science*, 45(1150):44, January 1917.

[LG94] X. Lagrange and P. Godlewski. Autonomous inter base station synchronisation via a common broadcast control channel. In *Proceedings of the 44th IEEE Vehicular Technology Conference (VTC 1994)*, pages 1050–1054, Stockholm, Sweden, June 1994.

[LGHD85] W.C. Lindsey, F. Ghazvinian, W.C. Hagmann, and K. Dessouky. Network synchronization. *Proceedings of the IEEE*, 73(10):1445–1467, October 1985.

[LSP82] L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, July 1982.

[LT91] W. Lewandowski and C. Thomas. GPS time transfer. *Proceedings of the IEEE*, 79(7):991–1000, July 1991.

[LW04] D. Lucarelli and I.-J. Wang. Decentralized synchronization protocols with nearest neighbor communication. In *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, pages 62–68, Baltimore, MD, USA, November 2004.

[MB02] M.S.O. Massunagaa and M. Bahiana. Synchronization in large populations of limit cycle oscillators with long-range interactions. *Physica D: Nonlinear Phenomena*, 168-169:136–141, August 2002.

[MBL03] H. Minn, V.K. Bhargava, and K.B. Letaief. A robust timing and frequency synchronization for OFDM systems. *IEEE Transactions on Wireless Communications*, 2(4):822–839, July 2003.

[Mei05] L. Meier. *Interval-Based Time Synchronization for Mobile Ad Hoc Networks*. PhD thesis, ETH Zurich, 2005.

[Mic] Microsoft. Windows Live Sync, http://sync.live.com, accessed on 18 december 2008.

[Mil90] D.L. Mills. On the accuracy and stability of clocks synchronized by the Network Time Protocol in the Internet system. *ACM Computer Communication Review*, 20(1):65–75, January 1990.

[Mil91] D.L. Mills. Internet time synchronization: the Network Time Protocol. *IEEE Transactions on Communications*, 39(10):1482–1493, October 1991.

[MKSL04] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi. The flooding time synchronization protocol. In *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, pages 39–49, Baltimore, MD, USA, November 2004.

[MM96] R. Mathar and J. Mattfeldt. Pulse-coupled decentral synchronization. *SIAM Journal on Applied Mathematics*, 56(4):1094–1106, August 1996.

[MM08] P.N. McGraw and M. Menzinger. Laplacian spectra as a diagnostic tool for network structure and dynamics. *Physical Review E*, 77(3):031102, March 2008.

[Mor05] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, February 2005.

[MS90] R.E. Mirollo and S.H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662, December 1990.

[New48] I. Newton. *Newton's Principia: The Mathematical Principles of Natural Philosophy*, chapter 6:Scholium, page 77. D. Adee, 1848.

[OHA07] P. Omiyi, H. Haas, and G. Auer. Analysis of TDD cellular interference mitigation using Busy-Bursts. *IEEE Transactions on Wireless Communications*, 6(7):2721–2731, July 2007.

[OM88] M. Oerder and H. Meyr. Digital filter and square timing recovery. *IEEE Transactions on Communications*, 36(5):605–612, May 1988.

[OS77] D. Otte and J. Smiley. Synchrony in Texas fireflies with a consideration of male interaction models. *Biology of Behaviour*, 2(2):143–158, 1977.

[OSM04] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, September 2004.

[PB05] C. Prehofer and C. Bettstetter. Self-organization in communication networks: Principles and design paradigms. *IEEE Communications Magazine*, 43(7):78–85, July 2005.

[PBA02] T. Pratt, C.W. Bostian, and J.E. Allnutt. *Satellite Communications*. Wiley, 2002.

[PDN07] A. Patel, J. Degesys, and R. Nagpal. Desynchronization: the theory of self-organizing algorithms for round-robin scheduling. In *Proceedings of the 1st IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007)*, pages 87–96, Boston, MA, USA, July 2007.

[Pes75] C.S. Peskin. *Mathematical Aspects of Heart Physiology*. New York: Courant Institute of Mathematical Sciences, 1975.

[PHS09] R. Pagliari, Y.-W.P. Hong, and A. Scaglione. A low-complexity algorithm for proportional fairness in body area networks. In *Proceedings of the 4th International Conference on Body Area Networks (BODYNETS 2009)*, Los Angeles, CA, USA, April 2009.

[PKR01] A. Pikovsky, J. Kurths, and M. Rosenblum. *Synchronization: A Universal Concept in Nonlinear Sciences*. Cambridge University Press, 2001.

[Rap01] T.S. Rappaport. *Wireless Communications: Principles & Practice*. Prentice Hall, 2nd edition, 2001.

[RF01] M.B.H. Rhouma and H. Frigui. Self-organization of pulse-coupled oscillators with application to clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):180–195, February 2001.

[RHK04] P. Rauschert, A. Honarbacht, and A. Kummert. On the IEEE 802.11 IBSS and its timer synchronization function in multi-hop ad hoc networks. In *Proceedings of the 1st International Symposium on Wireless Communication Systems (ISWCS 2004)*, pages 304–308, Mauritius, September 2004.

[Ric30] C.A. Richmond. Fireflies flashing in unison. *Science*, 71(1847):537–538, May 1930.

[Rob84] W.P. Robins. *Phase noise in signal sources: theory and applications*. Institution of Electrical Engineers, 2nd edition, 1984.

[Röm01] K. Römer. Time synchronization in ad hoc networks. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001)*, pages 173–183, Long Beach, CA, USA, October 2001.

[Röm05] K. Römer. *Time Synchronization and Localization in Sensor Networks*. PhD thesis, ETH Zurich, June 2005.

[SA05] W. Su and I.F. Akyildiz. Time-diffusion synchronization protocol for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 13(2):384–397, April 2005.

[SC97] T.M. Schmidl and D.C. Cox. Robust frequency and timing synchronization for OFDM. *IEEE Transactions on Communications*, 45(12):1613–1621, December 1997.

[SCS04] J.-P. Sheu, C.-M. Chao, and C.-W. Sun. A clock synchronization algorithm for multihop wireless ad hoc networks. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS 2004)*, pages 574–581, Tokyo, Japan, March 2004.

[SdV04] J.A. Stine and G. de Veciana. A paradigm for quality-of-service in wireless ad hoc networks using synchronous signaling and node states. *IEEE Journal on Selected Areas in Communications*, 22(7):1301–1321, September 2004.

[SN99] E. Sourour and M. Nakagawa. Mutual decentralized synchronization for intervehicle communications. *IEEE Transactions on Vehicular Technology*, 48(6):2015–2027, November 1999.

[SS93] S.H. Strogatz and I. Stewart. Coupled oscillators and biological synchronization. *Scientific American*, 269(6):68–74, December 1993.

[SSBNS08] O. Simeone, U. Spagnolini, Y. Bar-Ness, and S.H. Strogatz. Distributed synchronization in wireless networks. *IEEE Signal Processing Magazine*, 25(5):81–97, September 2008.

[STMA08] L. Sanguinetti, A. Tyrrell, M. Morelli, and G. Auer. On the performance of biologically-inspired slot synchronization in multicarrier ad hoc networks. In *Proceedings of the 67th IEEE Vehicular Technology Conference (VTC 2008-Spring)*, pages 21–25, Marina Bay, Singapore, May 2008.

[Str02] S.H. Strogatz. *Nonlinear Dynamics and Chaos*, chapter 4 - Flows on the Circle, pages 93–119. Westview, 2002.

[Str03] S.H. Strogatz. *Sync: The Emerging Science of Spontaneous Order*. Hyperion, 2003.

[SY04] F. Sivrikaya and B. Yener. Time synchronization in sensor networks: A survey. *IEEE Network*, 18(4):45–50, July 2004.

[TA98] F. Tong and Y. Akaiwa. Theoretical analysis of interbase-station synchronization systems. *IEEE Transactions on Communications*, 46(5):590–594, May 1998.

[TA07] A. Tyrrell and G. Auer. Imposing a reference timing on firefly synchronization in wireless networks. In *Proceedings of the 65th IEEE Vehicular Technology Conference (VTC 2007-Spring)*, pages 222–226, Dublin, Ireland, April 2007.

[TA08a] A. Tyrrell and G. Auer. Decentralized inter-base station synchronization inspired from nature. In *Proceedings of the 68th IEEE Vehicular Technology Conference (VTC 2008-Fall)*, pages 1–5, Calgary, Canada, September 2008.

[TA08b] A. Tyrrell and G. Auer. Decentralized slot synchronization for cellular mobile radio. *DoCoMo Technical Journal*, 10(1):60–67, June 2008.

[TA09] A. Tyrrell and G. Auer. Biologically inspired intercellular slot synchronization. *EURASIP Journal on Wireless Communications and Networking*, ID 854087:1–12, January 2009.

[TAB06a] A. Tyrrell, G. Auer, and C. Bettstetter. Fireflies as role models for synchronization in wireless networks. In *Proceedings of the 1st International Conference on Bio-Inspired Models of Network, Information and Computing Systems (BIONETICS 2006)*, pages 1–7, Cavalese, Italy, December 2006.

[TAB06b] A. Tyrrell, G. Auer, and C. Bettstetter. Synchronization inspired from nature for wireless meshed networks. In *Proceedings of the 2nd International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2006)*, pages 1–4, Wuhan, China, September 2006.

[TAB08a] A. Tyrrell, G. Auer, and C. Bettstetter. On the accuracy of firefly synchronization with delays. In *Proceedings of the 1st International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2008)*, pages 1–5, Aalborg, Denmark, October 2008.

[TAB08b] A. Tyrrell, G. Auer, and C. Bettstetter. A synchronization metric for meshed networks of pulse-coupled oscillators. In *Proceedings of the 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Systems (BIONETICS 2008)*, pages 1–7, Hyogo, Japan, December 2008.

[TAB09] A. Tyrrell, G. Auer, and C. Bettstetter. Emergent slot synchronization in wireless networks. *Accepted for publication in IEEE Transactions of Mobile Computing*, June 2009.

[TG06] S. Toumpis and A.J. Goldsmith. New media access protocols for wireless ad hoc networks based on cross-layer principles. *IEEE Transactions on Wireless Communications*, 5(8):2228–2241, August 2006.

[Tre01] H.L. Van Trees. *Detection, Estimation, And Modulation Theory: Part I*. Wiley, 2001.

[TWG02] M. Timme, F. Wolf, and T. Geisel. Prevalence of unstable attractors in networks of pulse-coupled oscillators. *Physical Review Letters*, 89(15):154105, September 2002.

[TWG04] M. Timme, F. Wolf, and T. Geisel. Topological speed limits to network synchronization. *Physical Review Letters*, 92(7):074101.1–074101.4, February 2004.

[TWM07] Y. Taniguchi, N. Wakamiya, and M. Murata. A traveling wave-based self-organizing communication mechanism for WSNs. In *Proceedings of the 5th ACM Conference on Embedded Networked Sensor Systems (SenSys 2007)*, pages 399–400, Sydney, Australia, November 2007.

[WATP+05] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal. Firefly-inspired sensor network synchronicity with realistic radio effects. In *Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys 2005)*, pages 142–153, San Diego, CA, USA, November 2005.

[WH00] B. Wegmann and M. Hellmann. Analysis of power control target levels in UTRA-TDD. In *Proceedings of the 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2000)*, pages 1216–1220, London , UK, September 2000.

[Win00] A.T. Winfree. *The Geometry of Biological Time*. Springer, 2nd edition, 2000.

[YHE02] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, pages 1567–1576, New York, NY, USA, June 2002.

[YS99] M.K.S. Yeung and S.H. Strogatz. Time delay in the kuramoto model of coupled oscillators. *Physical Review Letters*, 82(3):648–651, January 1999.

[ZL04] D. Zhou and T.-H. Lai. Analysis and implementation of scalable clock synchronization protocols in IEEE 802.11 ad hoc networks. In *Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004)*, pages 255–263, Fort Lauderdale, FL, USA, October 2004.

Short CV

EDUCATION

2005 - 2009	**DOCOMO Euro-Labs**, Munich, Germany **Klagenfurt University**, Klagenfurt, Austria Industrial PhD focusing on network slot synchronization inspired from the natural synchronization occurring in swarms of fireflies. Investigated PHY and MAC layer integration issues in wireless networks. Successfully integrated the scheme in the European project IST-WINNER.
2004 - 2005	**ENST Paris**, Paris, France Ecole Nationale Supérieure des Télécommunications. One year specialization courses leading to a Master of Research in Digital Communication Systems.
1999 - 2005	**ESIEE Paris**, Noisy-le-Grand, France Ecole Supérieure d'Ingénieurs en Electronique et Electrotechnique. Five-year programme leading to an equivalent of a Masters in Electrical Engineering. Majoring in Signal Processing and Telecommunications.
2003	**Chalmers University of Technology**, Gothenburg, Sweden Erasmus student following courses of the "Digital Communication Systems and Technology" Masters program during one semester.
1999	Baccalauréat Scientifique Option Mathématiques with honors Equivalent to a Secondary School diploma with minor in Mathematics.

EMPLOYMENT

2005	**DOCOMO Euro-Labs**, Munich, Germany Researched and developed a novel and robust slot synchronization scheme for self-organized networks based on the synchronization of fireflies.
2003 - 2004	**Texas Instruments**, Freising, Germany Gap year from university to gain professional experience. Provided technical support to European universities on TI Digital Signal Processors.
Summer 2002	**Asahi Denka Co. Ltd.**, Tokyo, Japan Analyzed chemical products in their largest factory and in their research center.
Summer 2001	**Banque BNP Paribas**, Paris, France Provided IT technical support on software and hardware issues for a user group of about $3,000$ people.
Summer 2000	**Beta Electronics**, Ashbourne, Co. Meath, Ireland Tested and corrected printed circuit boards in a factory which produces PCB used in video signal processing and in power transformers.