Michel RAYNAL

Member of Academia Europaea
Senior Member Institut Universitaire de France

Emeritus Professor, IRISA-ISTIC, Université de Rennes 1
Campus de Beaulieu, 35042 Rennes, France

Distinguished Chair Professor, Hong Kong Polytechnic University

https://team.inria.fr/wide/team/michel-raynal/
raynal@irisa.fr

February 2021

Part 3/3:

Past and Present Research Activities,

and List of Publications

- Previous part (1/3): CV, biography, and a view of my job in my scientific domain.
- Previous part (2/3): Recognition, management of scientific activities, and a few research grants in the recent past. Previous part (2/3):

To make the reading of each part as independent as possible from the other parts, a few short paragraphs appear in several parts.

Contents

1 General considerations

My research domain
The term distributed computing characterizes my research activity since 1983. The following list of keywords gives a a more detailed projection of my research.
Agreement problems (e.g., consensus), Asynchronous and synchronous systems, Causality, Checkpointing, Communication abstraction, Concurrent objects, Coordination, Distributed algorithms, Distributed computability, Dynamic systems, Fault-tolerance, Memory anonymity, Message-passing system, Process anonymity, Process coordination, Shared memory system, Symmetry breaking, Synchronization.

What is distributed computing? (Excerpt from the preface of my last book [1]).
Distributed computing was born in the late 1970s when researchers and practitioners started taking into account the intrinsic characteristic of physically distributed systems. The field then emerged as a specialized research area distinct from networking, operating systems, and parallel computing.

Distributed computing arises when one has to solve a problem in terms of distributed entities (usually called processors, nodes, processes, actors, agents, sensors, peers, etc.) such that each entity has only a partial knowledge of the many parameters involved in the problem that has to be solved. While parallel computing and real-time computing can be characterized, respectively, by the terms *efficiency* and *on-time computing*, distributed computing can be characterized by the term *uncertainty*. This uncertainty is created by asynchrony, multiplicity of control flows, absence of shared memory and global time, failure, dynamicity, mobility, etc. Mastering one form or another of uncertainty is pervasive in all distributed computing problems. A main difficulty in designing distributed algorithms comes from the fact that no entity cooperating in the achievement of a common goal can have an instantaneous knowledge of the current state of the other entities, it can only know their past local states.

Although distributed algorithms are often made up of a few lines, their behavior can be difficult to understand and their properties hard to state and prove. Hence, distributed computing is not only a fundamental topic but also a challenging topic where simplicity, elegance, and beauty are first-class citizens.

Defining a research domain is missing "something" if the definition is not accompagnied by references to the articles that "created" the research domain and established its foundations. As far as distributed computing is concerned, I consider the following articles.

- Lamport L., Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.
- Fischer M.J., Lynch N.A. and Paterson M.S., Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 32(2):374-382, 1985.
- Lamport L., On interprocess communication, Part 1: Basic formalism, Part II: Algorithms. *Distributed Computing*, 1(2):77-101, 1986.
- Herlihy M., Wait-free synchronization. *ACM Transactions on Programming Languages and Systems*, 13 (1):124-149, 1991.

Dissiminating knowledge: Books and surveys
I consider that writing books and surveys is important. Establishing new results or designing "disrupting" prototypes is fundamental but, from a society point of view, it is only one side of the coin. The other side consists in disseminating and transmitting the most important results to students and colleagues. While research transfer to industry impact the society in the short term, transfer of knowledge impact the society over the long term. This motivated me to participate in summer/winter schools and write books many surveys that appeared in journals (e.g., [R31,R43,R47,R65,R68,R72,R85,R87,R120,

[1] *Fault-tolerant message-passing distributed systems: an algorithmic approach*, Springer, 492 pages, 2018.

R153,R170,174]) or conference proceedings (e.g., [C123,C149,C150,C164, C201,C229,C234,266,283, 332,336]).

2 Scientific achievements: 1984-2000

I list and comment here only a a subset of my previous works. A reference of the type [Rx] refers to a journal paper, while a reference of the type [Cy] refers to a paper that appeared in a conference. In both cases the titles of the articles I consider as the most important appear in bold characters.

2.1 Early research

Very early research My early work was on operating systems and abstract data types. Then, I started working of communication systems with Gregor von Bochmann [R5]. It is during this "warm up" research period that I became interested in distributed computing.

Failure-free distributed synchronization My very first interest in distributed computing has been the mutual exclusion problem. I wrote a book on that topic and designed (with J.-M. Hélary and N. Plouzeau) one of the very first algorithms for arbitrary networks [R11]. A few years later, I designed (with Hélary and Mostéfaoui) a very general token-based mutex algorithm in which the token moves on an abstract tree that can be dynamically modified by an adversary daemon [R22]. (Interestingly, each of the mutex algorithms that uses a token moving a tree corresponds to a particular behavior of the underlying daemon.)

In the same spirit, I designed an algorithm for the h out of k resource allocation problem [C30]. While working on that topic, I entered the domain of quorums, and (with M. Mizuno) I introduced a general method to define and compose quorums [C33]. I also investigated the notion of k-arbiter [R39] to address the h out of k resource allocation problem.

Detection of stable/unstable properties My interest in the detection of stable properties started with my first PODC paper (1987) [C12] that presents a very general distributed detection algorithm for a large class of stable properties (properties on system global states). These properties are such that, once true, they remain true forever. Then, I addressed the case of algorithms targeted for specific stable properties, mainly distributed termination detection [C42] and distributed deadlock detection [R30].

Then, motivated by a project on distributed debugging, I started working on the detection of unstable properties. Here, the additional difficulty comes from the fact that such a property can be satisfied only intermittently. My main contributions to this topic are described in [R26,R28,R41,R48,C43,C46,C56,C66]. Among them, [R28] was one of the very first papers addressing properties defined on the many control flows present in a distributed execution, while [R41] introduced the notion of *inevitable global state* and defined an algorithm to detect them (such a state is a state that is seen by all the sequential observers of the corresponding distributed execution). Finally the work described in [R48] concerns the detection of conjunction of local predicates; it presents one of the most efficient algorithms proposed so far to detect on the fly such predicates.

Data consistency Distributed computing involves distributed data. This part of my work has many facets. In [R51], Vijay Garg and I introduced the *normality* consistency condition. Its definition is based only on the local order of operations as perceived by each process and by each object. If each operation is on exactly one object, normality and linearizability[2] are the same. Differently, when operations span

[2]Herlihy M.P. and Wing J.M., Linearizability: a Correctness Condition for Concurrent Objects. *ACM Transactions on Programming Languages and Systems*, 12(3):463-492, 1990.

several objects, normality is weaker than linearizability.

I have also proposed several protocols to implements sequentially consistent memories [R74,C47,C54] and shown that sequential consistency can be seen as a for of lazy linearizability [C148].

I also investigated (mainly with M. Ahamad from Georgia Tech) the notion of *timed consistency* for shared distributed objects. This work was published in the top conferences PODC and DISC [C97,C100] and in journals (e.g., [R71]).

2.2 Causality, checkpointing, virtual precedence and vector clocks

Causal order An important part of my past work was on *causality* in message-passing systems and its applications. This fundamental notion was introduced in 1978 by L. Lamport[3].

One of my early work on causality is a protocol to deliver messages according to the so-called *causal order*. This simple and elegant protocol (designed with A. Schiper and S. Toueg), that appeared in IPL [R17], is widely referenced and appears in several textbooks devoted to distributed computing.

Checkpointing Then, my interest in causality moved to the checkpointing problem, and more specifically to communication-induced checkpointing (CIC). This checkpointing technique allows the application messages to piggyback control information, but does not allow the use of additional control messages. In this context, We (I with mainly J.-M. Hélary, A. Mostéfaoui) produced several results, among which the following ones.

- An important theoretical question is "Given an arbitrary set of local checkpoints, do these local checkpoints belong to the same consistent global checkpoint?" While this question has been answered by Netzer and Xu in the particular context of message passing systems[4], we answered it in a very general asynchronous computational model that encompasses shared memory systems and various message passing systems with reliable or unreliable and point-to-point or multicast or broadcast communication. This result has been published in *Acta Informatica* [R45].
- A very general definition of global checkpoint consistency that appeared in *IEEE Transactions on Software Engineering* [R49].
- A communication-induced snapshot algorithm that appeared in *IEEE TPDS* [R53]. This very generic algorithm can be instantiated in many ways. It shows that consistent global states can be determined without the help of additional control messages (differently from the well-known Chandy and Lamport's snapshot algorithm which uses additional control messages[5]).
- A family of algorithms that allows the processes to define independent local checkpoints in such a way that any local checkpoint is part of a consistent global checkpoint. This work appeared in *Distributed Computing* [R57].
- The impossibility to design scalar-based communication-induced checkpointing protocols that satisfy the Rollback-Dependency Trackability property. This work appeared in *Information processing Letters* [R61].
- A Characterization of the *Rollback-Dependency Trackability* property. This work appeared in *Information and Computation* [R62].

All this work participated in providing checkpointing with solid theoretical foundations.

---

[3] Lamport L., Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.

[4] Netzer, R.H.B. and Xu, J., Necessary and Sufficient Conditions for Consistent Global Snapshots, *IEEE Transactions on Parallel and Distributed Systems*, 6(2):165-169, 1995.

[5] Chandy K.M. and Lamport L., Distributed snapshots: determining global states of distributed systems, *ACM Transactions on Computer Systems*, 3(1):63–75, 1985.

Virtual precedence In [R70,C87], Hélary, Mostéfaoui, and I introduced and investigated the concept of *virtual precedence*. The problem is the following. An interval of a sequential process is a sequence of consecutive events of this process. The set of intervals defined on a distributed computation defines an abstraction of this distributed computation, and the traditional causality relation on events induces a relation on the set of intervals. The question is then: "Is the interval-based abstraction associated with a distributed computation consistent?". To answer this question, this paper introduces the *Interval Consistency* (IC) condition. Intuitively, this condition states that an interval-based abstraction of a distributed computation is consistent if its precedence relation does not contradict the sequentiality of each process. Interestingly, the IC condition can be operationally characterized in terms of timestamps (whose values belong to a lattice). The paper uses this characterization to design a versatile protocol that, given intervals defined by a daemon whose behavior is unpredictable, breaks them (in a non trivial manner) in order to produce an abstraction satisfying the IC condition. (Among other problems, communication-induced checkpointing can benefit from IC.)

Optimal implementation of vector clocks The major parts of the previous protocols are based on vector clocks, whose size is equal to the number of processes. So, an important question is the following "Given a message m, which is the minimal subset of the entries of a vector clock that m has to piggyback in order to fully capture causality?" We answered this question by stating a necessary and sufficient condition, and showed how it can implemented. This is an important result as, for each message m, it states the minimal quantity of information m has to carry in order the vector clock system allows the processes to fully capture the causality relation [R57]. Moreover, we also designed the first (to our knowledge) distributed algorithm that computes on the fly the *transitive reduction* associated with the partial order defined by a distributed execution [R73,C144] (the transitive reduction captures exactly the minimal partial order associated with a distributed execution).

3 Scientific achievements: 2000-2010

Since more than 20 years, an important part of my research is mainly focused on algorithms for distributed agreement, with a few incursions into distributed computability.

One of my first work in distributed agreement is a (co-authored) paper titled "From group communications to transactions in distributed systems" that appeared in a 1996 special issue of *Communications of the ACM* devoted to "group communication" [R36]. Then, I became interested in the consensus problem, its variants such as the k-set agreement, and the design of protocols implementing failure detectors.

3.1 Consensus

I designed several consensus algorithms in asynchronous message-passing systems or shared memory systems for the crash failure model [R54,R67,R69,R82,R83,R95,C110,C134] or the Byzantine failure model [R79,R89,R90].

More specifically, [C110] (DISC 1999) presents a very simple consensus algorithm that is generic in the sense that it can be instantiated with any failure detector of any failure detector class as defined by Chandra and Toueg[6]. This algorithm was the first failure detector-based consensus algorithm that uses quorums in an explicit way to ensure the agreement property. Interestingly, the pattern on which this algorithm is based is very similar to the *adopt/commit* pattern proposed by Gafni[7]. Many consensus algorithms proposed after ours, follows our pattern. Similarly, the leader-based consensus protocol

[6]Chandra T.D. and Toueg S., Unreliable Failure Detectors for Reliable Distributed Systems. *Journal of the ACM*, 43(2):225-267, 1996.

[7]Gafni E., Round-by-round Fault Detectors: Unifying Synchrony and Asynchrony. *Proc. 17th ACM Symposium on Principles of Distributed Computing (PODC'00)*, ACM Press, pp. 143-152, 1998.

described in [R67], that was one of the very first Omega-based protocol, is widely referenced in the literature.

I have been one of the very first researcher to investigate the situations where consensus can be solved in one communication step. The corresponding paper [C134] is also widely referenced. In the same spirit, a transformation from binary consensus to multi-valued consensus is presented in [R58].

I revisited (with Y. Moses) the simultaneous consensus problem [C228] and showed that it cannot benefit from the condition-based approach. "Simultaneous" means that the processes that decide have to decide during the very same round. I also investigated the use of consensus to solve agreement problems such as total order multicast in overlapping groups [R60], atomic broadcast in crash/recovery systems whose aim is to implement quorum-based replication [R77], and how to save consensus executions when one has to implement atomic broadcast [C124].

3.2 Failure detectors

I have addressed several directions of research related to failure detectors.

Computing with failure detectors In [R59,R76] (with co-authors) I investigated the *Global Data Computation* problem (also known under the name *Interactive Consistency*[8]). It consists in providing each process with the same vector (with one entry per process) such that each entry is filled with the value supplied by the corresponding process if it has not crashed, and its value or $\perp$ otherwise. The paper [R76] presents an algorithm, based on a perfect failure detector, that requires the processes to execute at most $\min(f+2, t+1)$ rounds (where t is the model upper bound on the number of processes that can crash, and f the actual number of crashes). This showed that solving that problem in an asynchronous system enriched with a perfect failure detector is not more expensive than solving it in a synchronous system.

From one failure detector to another one With A. Mostéfaoui, I introduced the notion of failure detectors with *limited scope accuracy* [C111,C121] (they are failure detectors whose scope is limited to a subset of $x \leq n$ processes). [R81] presents a necessary and sufficient condition that allows transforming a failure detector with a limited scope accuracy into its non-limited scope counterpart. This condition states that the scope x of the failure detector has to be greater than t (the upper bound on the number of processes that can crash). Other failure detector transformations are described in [R56] and [R94]. This last JPDC paper presents a transformation that is quiescent (after some finite time, the transformation does not require message to be exchanged).

On the weakest failure detector classes [R97] shows that there is no one-shot agreement problem for which the failure detector class $\Diamond\mathcal{P}$ is the weakest that allows to solve it (the class $\Diamond\mathcal{P}$ is the class of *eventually perfect* failure detectors: after some finite time they do suspect only the processes that have crashed). The articles [R103,C198] study classes of failure detectors whose power can be added to conditions to obtain an "arithmetic" of failure detector classes.

Implementing failure detectors Traditional implementations of failure detectors consider that the system satisfy additional synchrony assumptions. In [C160,R93] I proposed (with Mostéfaoui and Mourgaya) a totally new approach to implement failure detectors. That approach relies on the pattern of messages exchanged by the processes. Basically, this novel assumption requires that there is one (or several) process(es) whose answers to queries are among the $n-t$ first answers received by the

[8]Pease L., Shostak R. and Lamport L., Reaching Agreement in Presence of Faults. *Journal of the ACM*, 27(2):228-234, 1980.

process that issued the query. I have then shown that this novel kind of assumption can be combined with timing assumptions associated with a subset of the channels [R92]. It is important to see that such an approach favors the assumption coverage.

In [R113] is presented a weak timing assumption that allows the construction of an eventual leader service (Omega) in an asynchronous shared memory system, while [C221] considers the case of a message-passing system. Unifying these two approaches is still an open problem.

3.3 The condition-based approach

The consensus problem is one of the most important problem in asynchronous systems prone to failures: it captures the difficulty of coordinating independent entities when these entities can fail. Each process proposes a value, and all the non-faulty processes have to agree on the same value that has to be one of the proposed values. Unfortunately, despite its very simple formulation, there is no deterministic algorithm that can solve the consensus problem as soon as (even only) one process can crash. This is the famous FLP impossibility result[9]. Until 2000, mainly three approaches were proposed to circumvent this impossibility. One consists in looking for a non-deterministic solution: a process can draw random numbers[10]. Another consists in enriching the system with a failure detector of an appropriate class[6]. The last one, that concerns shared-memory systems, consists in enriching the system with operations stronger than the base read and write operations (e.g. a Compare&Swap() operation). This approach has given rise to the notion of consensus number and to Herlihy's hierarchy[11].

In 2000, with S. Rajsbaum and A. Mostéfaoui (during a visit of S. Rajsbaum at IRISA) we started a collaboration that proved to be very fruitful, namely we introduced a novel way of circumventing the FLP impossibility result. This new approach does not consist in enriching the underlying system, but in restricting the set of input vectors (such a restricted set defines a *condition*). The main challenge is then to characterize the largest conditions that allows solving consensus in presence of up to t process crashes. This is the problem solved in [R78]. Then, [R80] establishes a hierarchy of conditions and presents corresponding efficient algorithms.

While the result in [R78] establishes the frontier from which a condition allows solving consensus despite asynchrony and up to t process crashes, we showed in [R91] that a condition can allow expediting consensus in a synchronous system. Said differently, [R91] extends to synchronous systems the hierarchy of conditions, namely, a condition that allows solving consensus despite t crashes in an asynchronous system, allows solving consensus optimally (in two rounds) in a synchronous systems: decidability in asynchronous system can be converted in efficiency in a synchronous system.

In a very interesting way, we have shown that there is a very strong connection between error-correcting code and distributed agreement problems [R98] (intuitively, an input vector encodes a value that has to be decided by the processes). We have also investigated the combined power of conditions and failure detectors to solve agreement problems [R105]. With my PhD student F. Bonnet, I also extended the condition-based approach to address the k-set agreement problem [R110]. This provides us with a very general condition-based framework.

[9] Fischer M.J., Lynch N.A. and Paterson M.S., Impossibility of Distributed Consensus with One Faulty Process. *Journal of the ACM*, 32(2):374-382, 1985.

[10] Ben-Or M., Another Advantage of Free Choice: Completely Asynchronous Agreement Protocols. *Proc. 2nd ACM Symposium on Principles of Distributed Computing (PODC'83)*, pp. 27-30, 1983.
Rabin M., Randomized Byzantine Generals. *Proc. 24th IEEE Symposium on Foundations of Computer Science (FOCS'83)*, IEEE Computer Press, pp. 403-409, 1983.

[11] Herlihy M.P., Wait-Free Synchronization. *ACM Transactions on Programming Languages and Systems*, 13(1):124-149, 1991.

3.4 k-Set agreement in synchronous/asynchronous systems

The k-set agreement problem is a weakening of the consensus in the sense that up to k different values can be decided. I worked on this problem in failure prone synchronous and asynchronous systems.

Set agreement in synchronous systems The main result in this type of systems is described in [R112], where the fault model is the general omission failure model. A new termination property is introduced. Let a *good* process be process that neither crashes nor commits receive omission failure. The new property, called *strong termination*, obliges the good processes to decide. A k-set algorithm is presented where every good process decides and halts by round $\min(\lfloor \frac{f}{k} \rfloor + 2, \lfloor \frac{t}{k} \rfloor + 1)$. This algorithm is clearly optimal. In addition to the strong termination property, (as far as I know) this algorithm is the only early-deciding and stopping k-set agreement algorithm proposed so far for the general omission failure. A survey of synchronous set agreement is presented in [201].

Set agreement in asynchronous systems Differently from synchronous systems, k-set agreement cannot be solved in asynchronous systems when $k \leq t$. As already indicated, [C121] investigates failure detectors with limited scope accuracy to solve k-set agreement, and presented several protocols. Differently, [C131] considers that the additional power is supplied by random numbers. [R105,C184] add the power of appropriate failure detector classes to conditions in order to solve k-set agreement. In the context of shared memory systems, the invited paper [C205] posed the problem of the weakest failure detector for solving the k-set agreement problem. At the rump session of PODC 2007, I posed the following conjecture at the rump session "$\overline{\Omega_k}$ *is the weakest failure detector for* $k-$*set agreement in shared memory systems*". This conjecture has been positively answered by three independent groups of researchers (PODC 2009). I am now working for the weakest failure detector for $k-$set agreement in message passing systems.

3.5 Distributed computability

I started becoming interested in distributed computability in 2006. I worked on this topic mainly with E. Gafni, S. Rajsbaum, and C. Travers (who, at that time, was one of my PhD students).

Committee decision and renaming My first (co-authored) contribution was the definition of the *committee decision* problem [C192] that was later generalized to the notion of *simultaneous consensus task* [C204].The question answered is the following: "What is the power of the task where processes are involved in k simultaneous consensus instances and each is required to decide in only one of them (several processes deciding possibly in different instances)?"

My second contribution is a wait-free k-Test&Set-based adaptive renaming algorithm [C200]. The paper shows that the new space name can be reduced from $2p-1$ (where p is the number of participating processes) to $2p - \lceil \frac{p}{k} \rceil$. This work encouraged us to investigate the relation linking Test&Set, adaptive agreement and set agreement. This investigation resulted in [C216,C223]. As Test&Set, adaptive agreement and set agreement are sub-consensus tasks (i.e., they are weaker than consensus), the holy grail is here to establish a hierarchy of subconsensus tasks (if any), similarly to the hierarchy of consensus tasks as defined by Herlihy[11]. This line of research is fundamental if we want to understand the power of base synchronization primitives. In some sense, the research agenda is here to establish the "Mendeleiev's table" of sub-consensus tasks. A very early step in that direction appears in [R106].

I also proposed and investigated (with G. Taubenfeld) the notion of *timed register* [C208]. A timed register generalizes the notion of an atomic register as follows: if a process invokes two consecutive operations on the same timed register which are a read followed by a write, then the write operation is executed only if it is invoked at most d time units after the read operation, where d is defined as part of the read operation. We show that a timed register is a universal object (i.e., an object from which any object defined by a sequential specification can be built despite asynchrony and failures)[11].

[C223] explores a new direction to solve the k-set agreement problem in a synchronous system of n processes. It considers that the system is enriched with base objects (denoted $[m, \ell]$_SA objects) that allow solving the ℓ-set agreement problem in a set of m processes ($m < n$). This work has several contributions. It first proposes a synchronous k-set agreement algorithm that benefits from such underlying base objects. This algorithm requires $O(\frac{t\ell}{mk})$ rounds, more precisely, $\lfloor \frac{t}{\Delta} \rfloor + 1$ rounds, where $\Delta = m\lfloor \frac{k}{\ell} \rfloor + (k \bmod \ell)$. It also shows that this bound, that involves all the parameters that characterize both the problem (k) and its environment (t, m and ℓ), is a lower bound. This work is then extended to the early deciding case. It presents a k-set agreement algorithm that directs the processes to decide and stop by round $\min\left(\lfloor \frac{f}{\Delta} \rfloor + 2, \lfloor \frac{t}{\Delta} \rfloor + 1\right)$. These bounds generalize the bounds previously established for solving the k-set agreement problem in pure synchronous systems.

Formal models for distributed computing The *Iterated Immediate Snapshot* (IIS) model has been introduced by E. Borowsky and E. Gafni[12]. It is an asynchronous computation model where processes communicate through a sequence of one-shot *immediate snapshot* objects. It is known that this model is equivalent to the usual asynchronous read/write shared memory model, for wait-free task solvability. Its interest lies in the fact that its runs are more structured and easier to analyze than the runs in the shared memory model. As the IIS model and the shared memory model are equivalent for wait-free task solvability, a natural question is the following: "Are they still equivalent for wait-free task solvability, when they are enriched with the same failure detector?" Rajsbaum, Travers, and I showed in [R100] that the answer to this question is "no."

At first glance, this answer can appear counter-intuitive. So, the next question is the following: "Given a shared memory model enriched with a failure detector, what is an equivalent IIS model?" [C226] shows that an elegant way of capturing the power of a failure detector and other partially synchronous systems in the IIS model is by restricting appropriately its set of runs, giving rise to the *Iterated Restricted Immediate Snapshot* model (IRIS).

3.6 Software transactional memory and synchronization

I started becoming interested in software transactional memories (STM) in 2007. Basically, the problem consists in discharging the application programmer from the management of the underlying synchronization in multiprocess programs that access shared objects. From a practical point of view, STM systems are one of the most promising approach to take up challenge posed by the recent advance of multicore architectures and the deployment of multiprocessors as the mainstream computing platforms. From a theoretical point of view, STM systems give a new impetus that forces to rethink the way synchronization problems have to be solved (basically, synchronization is coming back, but it is not the same [C224]).

My initial contributions concern the following points. While the fate of a transaction is to commit or abort, no current STM specification states situations where the STM system is forced to commit a transaction. In [C231], with my PhD student D. Imbs, I introduced a new property called *obligation* that specifies situations where a transaction cannot be aborted. A corresponding STM protocol is presented and formally proved correct.

The second contribution is the definition of a general framework to state consistency conditions suited to STM systems [C236]. This framework not only encompasses serializability, strict serializability and opacity (that are "traditional" consistency conditions), but permits to define new meaningful consistency conditions. Among them the (new) *virtual world* condition is particularly interesting: it is

[12] Borowsky E. and Gafni E., Immediate Atomic Snapshots and Fast Renaming. *Proc. 12th Principles of Distributed Computing (PODC'93)*, ACM Press, pp. 41-51, 1993.

less restrictive than than opacity, while requiring that (whatever the fact that it commits or aborts) any transaction always reads its values from a consistent global state.

4 Scientific achievements: 2010-2015

Whatever their granularity, today distributed applications are pervasive and benefit everyone (e.g., P2P, cloud computing, sensors networks, or social networks for "large grain" applications, and multicore for "small grain" applications). All these applications are becoming larger and larger and more and more distributed. The development of such platforms and their usage have somehow preceded their theoretical foundations. Up to now, their design principles look sometimes more like a "trick" than well-mastered basic principles. The explosion of the number of distributed applications and the number of "computing adversaries" such as scaling, misbehaviors (also characterized as malicious behaviour when referring to entities attempting to voluntarily or not hurt the system), dynamicity, etc., makes their basic principles more and more difficult to grasp. Traditional algorithms simply do not fit this challenging new setting and it is required to revisit the field.

Hence, research addressing distributed computing theory that can benefit future applications is more needed than ever. This is a great challenge for the computer science community and constitutes my main research motivation since 2010. As already said, of my leitmotifs is "When something works we have to know why it works, and when it does not work we have to know why it does not work".

4.1 Symmetry breaking

Symmetry breaking (in the presence of failures) is a fundamental problem of distributed computing. More precisely, processes in a concurrent system need to coordinate using an underlying shared memory or a message-passing system in order to solve agreement tasks such as, for example, consensus or set agreement. However, coordination is often needed to break the symmetry of processes that are initially in the same state, for example, to get exclusive access to a shared resource, to get distinct names, or to elect a leader.

I (with A. Castañeda, D. Imbs, and S. Rajsbaum) introduced and studied the family of generalized symmetry breaking (GSB) tasks, that includes election, renaming and many other symmetry breaking tasks [C265]. Differently from agreement tasks, a GSB task is inputless, in the sense that processes do not propose values; the task only specifies the symmetry breaking requirement, independently of the initial state of the system (where processes differ only on their identifiers). Among various results characterizing the family of GSB tasks, we showed is shown that perfect renaming is universal for all GSB tasks [C275].

We then studied the power of renaming with respect to k-set agreement. We showed that, in a system of n processes, perfect renaming is strictly stronger than $(n-1)$-set agreement, but not stronger than $(n-2)$-set agreement. Furthermore, $(n+1)$-renaming cannot solve even $(n-1)$-set agreement. As a consequence, there are cases where set agreement and renaming are incomparable when looking at their power to implement each other. We also showed that there is a large family of GSB tasks that are more powerful than $(n-1)$-set agreement [C287]. Some of these tasks are equivalent to n-renaming, while others lie strictly between n-renaming and $(n+1)$-renaming. Moreover, none of these GSB tasks can solve $(n-2)$-set agreement. Hence, the GSB tasks have a rich structure and are interesting in their own. The proofs of these results are based on combinatorial topology techniques and new ideas about different notions of non-determinism that can be associated with shared objects. Interestingly, this paper sheds a new light on the relations linking set agreement and symmetry breaking. All these results are pieced together in [R145]. We also showed that the notion of a *process group* allows the renaming space to be reduced according to the number of process groups [C284].

4.2 Failure detectors

Hybrid distributed system With my PhD student D. Imbs, I introduced an asynchronous crash-prone hybrid system model, where the system is hybrid in the way the processes can communicate. On the one side, a process can send messages to any other process. On another side, the processes are partitioned into clusters and each cluster has its own read/write shared memory. In addition to the model, one of our contributions concerns the implementation of an atomic register in this system model. More precisely, we introduced a new failure detector (denoted $M\Sigma$) and showed that, when considering the information on failures needed to implement a register, this failure detector is the weakest. To that end, we presented an $M\Sigma$-based algorithm that builds a register in the considered hybrid system model and showed that it is possible to extract $M\Sigma$ from any failure detector-based algorithm that implements a register in this model. We also (a) showed that $M\Sigma$ is strictly weaker than Σ (which is the weakest failure detector to implement a register in a classical message-passing system) and (b) presented a necessary and sufficient condition to implement $M\Sigma$ in a hybrid asynchronous communication system. These result are described in [R134] and [C269] (which obtained the Best Paper Award at SSS 2011.

Iterated distributed model The basic distributed asynchronous read/write computation model is made up of n asynchronous processes which communicate by reading and writing atomic registers only. The distributed asynchronous iterated model is a more constrained model in which the processes execute an infinite number of rounds and communicate at each round with a new object called immediate snapshot object. Moreover, in both models up to $n - 1$ processes may crash in an unexpected way. When considering computability issues, two main results are associated with the previous models. The first states that they are computationally equivalent for decision tasks. The second states that they are no longer equivalent when both are enriched with the same failure detector.

With my PhD student J. Steiner, I showed how to capture failure detectors in each model so that both models become computationally equivalent [C280]. To this end, I introduced the notion of a "strongly correct" process which appears particularly well-suited to the iterated model, and presents simulations that prove the computational equivalence when both models are enriched with the same failure detector. I also extended also these simulations to the case where the wait-freedom requirement is replaced by the notion of t-resilience. (The important new idea is here the notion of strongly correct processes. Those are the processes that "see" each other infinitely often. If, after some time, a process is late, it sees the values written by the strongly correct processes but the values it writes are not seen by them.)

4.3 Process anonymity

Due to the multiplicity of loci of control, a main issue distributed systems have to cope with lies in the uncertainty on the system state created by the adversaries that are asynchrony, failures, dynamicity, mobility, etc. Considering message-passing systems, I addressed (with my PhD student F. Bonnet) the uncertainty created by the net effect of asynchrony and process crash failures in systems where the processes are anonymous (i.e., processes have no identity and locally execute the very same algorithm).

Trivially, agreement problems such as consensus, that cannot be solved in non-anonymous asynchronous systems prone to process failures, cannot be solved either if the system is anonymous. So, we investigated failure detectors that allow processes to circumvent this impossibility.

In [R121] we introduced a failure detector class denoted ψ, that gives to each process an upper bound on the number of processes that are currently alive (in a non-anonymous system, the classes ψ and $\mathcal{P}$ -the class of perfect failure detectors- are equivalent). We designed a simple ψ-based consensus algorithm where the processes decide in $2t + 1$ asynchronous rounds (where t is an upper bound on the number of faulty processes), and showed shows that $2t + 1$ is a lower bound for consensus in the anonymous systems equipped with ψ. We then showed addressed early-decision, and presented and proved an early-deciding algorithm where the processes decide in $\min(2f + 2, 2t + 1)$ asynchronous rounds (where f is the actual number of process failures). This leads to think that anonymity doubles the cost (wrt synchronous systems) and it is conjectured that $\min(2f+2, 2t+1)$ is the corresponding lower bound.

We then continued our work of anonymous distributed systems and presented four failure detectors (denoted AP, $\overline{AP}$, $A\Omega$, and $A\Sigma$) and show that they are the "identity-free" counterparts of perfect failure detectors, eventual leader failure detectors and quorum failure detectors, respectively. $A\Sigma$ is new and showing that $A\Sigma$ and Σ have the same computability power in a non-anonymous system is not trivial. We also showed that the notion of failure detector reduction is related to the computation model. Then, we presented and proved correct a uniform anonymous consensus algorithm based on the failure detector pair $(A\Omega, A\Sigma)$ ("uniform" means here that not only processes have no identity, but no process is aware of the total number of processes). This new algorithm is not a simple "straightforward extension" of an algorithm designed for non-anonymous systems. To benefit from $A\Sigma$, it uses a novel message exchange pattern where each phase of every round is made up of sub-rounds in which appropriate control information is exchanged. Finally, we introduced the notions of failure detector hierarchy, weakest failure detector for anonymous consensus, and the implementation of identity-free failure detectors in anonymous systems. This work was published in [R133, C254].

I also studied the computability power of homonymous systems in [C277]. In these systems, several processes can have the same name. If all processes have distinct names we are in a classical distributed system, while we are in an anonymous system if all the processes have the same name. I mainly studied the solvability of the consensus problem in such a context.

4.4 Synchronous vs asynchronous systems: when are they equivalent?

A message adversary is a daemon that suppresses messages in round-based message-passing synchronous systems in which no process crashes. This notion has first been introduced by N. Santoro and P. Widmayer a long time ago[13]. A property imposed on a message adversary defines a subset of messages that cannot be eliminated by the adversary. It has recently been shown[14] that when a message adversary is constrained by a property denoted TOUR (for tournament), the corresponding synchronous system and the asynchronous crash-prone read/write system have the same computability power for task solvability.

We my PhD student J. Stainer, I investigated in [C290] new message adversary properties (denoted SOURCE and QUORUM), and showed that the synchronous round-based systems whose adversaries are constrained by these properties are characterizations of classical asynchronous crash-prone systems (1) in which processes communicate through atomic read/write registers or point-to-point message-passing, and (2) enriched with failure detectors such as the eventual leader Ω and the quorum failure detector Σ. Hence these properties characterize maximal adversaries, in the sense that they define strongest message adversaries equating classical asynchronous crash-prone systems. They consequently provide strong relations linking round-based synchrony weakened by message adversaries with asynchrony restricted with failure detectors. This not only enriches our understanding of the synchrony/asynchrony duality, but also allows for the establishment of a meaningful hierarchy of property-constrained message adversaries.

4.5 Asynchronous systems with Byzantine processes

Since 2013, I started to work again on Byzantine failures (a process commits a Byzantine failure when it behaves arbitrarily). My main results concern two of the most important problems of message-passing

[13] N. Santoro and P. Widmayer, Time is not a healer, *Proc. 6th Annual Symposium on Theoretical Aspects of Computer Science (STACS'89)*, Springer LNCS 349, pp. 304-316, 1989. This notion is also addressed by Kuhn F., Lynch N.A., and Oshman R.,in their paper "Distributed computation in dynamic networks", *Proc. 42nd ACM Symposium on Theory of Computing (STOC'10)*, ACM press, pp. 513-522, 2010.

[14] Afek Y. and Gafni E., Asynchrony from synchrony, *Proc. Int'l Conference on Distributed Computing and Networking (ICDCN'13)*, Springer LNCS 7730, pp. 225-239, 2013.

distributed computing in the presence of failures, namely, agreement (consensus) and the construction of a read/write shared memory abstraction.

Optimal consensus in the presence of Byzantine processes With my colleague A. Mostéfaoui, I considered the consensus problem in asynchronous message-passing systems. We designed a new round-based asynchronous consensus algorithm that copes with up to $t < n/3$ Byzantine processes, where n is the total number of processes. In addition of not using signature, not assuming a computationally-limited adversary, while being optimal with respect to the value of t, this algorithm has several noteworthy properties: the expected number of rounds to decide is four, each round is composed of two or three communication steps and involves $O(n^2)$ messages, and a message is composed of a round number plus a single bit. To attain this goal, the consensus algorithm relies on a common coin as defined by Rabin, and a new extremely simple and powerful broadcast abstraction suited to binary values. The main target when designing this algorithm was to obtain a cheap and simple algorithm. This was motivated by the fact that, among the first-class properties, simplicity –albeit sometimes under-estimated or even ignored– is a major one. The paper presenting this result [C301] obtained the Best Paper Award at the ACM Conference on Principles of Distributed Systems (PODC 2014), which is the premier conference in the domain. A journal version appeared in JACM [R140], I also showed that multivalued consensus can be solved with $O(n^2)$ expected messages [R155].

Building a shared memory on top of a Byzantine message-passing system With D. Imbs, S. Rajsbaum, and J. Stainer, I addressed the construction and the use of a shared memory abstraction on top of an asynchronous message-passing system in which up to t processes may commit Byzantine failures [C303]. This abstraction consists of arrays of n single-writer/multi-reader atomic registers, where n is the number of processes. A distributed algorithm building such a shared memory abstraction it first presented. This algorithm assumes $t < n/3$, which is shown to be a necessary and sufficient condition for such a construction. Hence, the algorithm is resilient-optimal. Then we presented a distributed algorithms built on top of this shared memory abstraction, which cope with up to t Byzantine processes. The simplicity of these algorithms constitutes a strong motivation for such a shared memory abstraction in the presence of Byzantine processes.

For a lot of problems, algorithms are more difficult to design and prove correct in a message-passing system than in a shared memory system. Using a protocol stacking methodology, the aim of the proposed abstraction is to allow an easier design (and proof) of distributed algorithms, when the underlying system is an asynchronous message-passing system prone to Byzantine failures.

4.6 Concurrent data structures

An atomic snapshot object is an object that can be concurrently accessed by asynchronous processes prone to crash. It is a fundamental object of concurrent programming in the presence of failures[15]. It is made of m components (base atomic registers) and is defined by two operations: an update operation that allows a process to atomically assign a new value to a component, and a snapshot operation that atomically reads and returns the values of all the components.

In [R127] I proposed an algorithm implementing a partial snapshot object, i.e., an object where the snapshot operation that can take any subset of the components as input parameter, and atomically reads and returns the values of this subset of components. This algorithm is based on new notions called called *help-locality* and *freshness*. Help-locality requires that an update operation helps only the concurrent partial snapshot operations that read the component it writes. When an update of a component r helps a partial snapshot, freshness requires that the update provides the partial snapshot

[15] Afek Y., Attiya H., Dolev D., Gafni E., Merritt M. and Shavit N., Atomic Snapshots of Shared Memory. *Journal of the ACM*, 40(4):873-890, 1993.

with a value of the component r that is at least as recent as the value it writes into that component. (No snapshot algorithm proposed so far satisfies these properties). The algorithm is wait-free, linearizable and satisfies the previous efficiency properties. Interestingly, the principle that underlies the proposed algorithm is different from the one used so far, namely, it is based on the "write first, and help later" strategy. An improvement of the previous algorithm, based on LL/SC atomic registers, is also presented, which decreases the number of base registers from $O(n^2)$ to $O(n)$.

With T. Crain (Marie Curie PhD student) we investigated efficient implementations of concurrent data structures such as binary trees, skip-lists, etc. The corresponding results are described in [C273,C288,C291].

4.7 Software transactional memory again

The aim of a Software Transactional Memory (STM) system is to discharge the programmer from the explicit management of synchronization issues. The programmer's job resides in the design of multiprocess programs in which processes are made up of transactions, each transaction being an atomic execution unit that accesses concurrent objects. The important point is that the programmer has to focus her/his efforts only on the parts of code which have to be atomic execution units without worrying on the way the corresponding synchronization has to be realized.

After having introduced the *virtual world consistency* (VWC) condition [R128], I have shown that the three properties read invisibility, permissiveness and opacity are incompatible, while read invisibility, permissiveness and VWC are compatible [C271]. While opacity requires that all the transactions (be them aborted or committed) appear as being totally ordered, VWC is weaker in as it only requires that an aborted transaction be ordered with respect to committed transactions only. This allows more transactions to be committed.

Among other results I also designed (with my PhD students D. Imbs and T. Crain) a universal construction for transaction-based multiprocess systems [R135]. This construction is such that (1) every invocation of a transaction is executed exactly once and (2) the notion of commit/abort of a transaction remains unknown to the programmer. This system, which imposes restriction neither on the design of processes nor on their concurrency pattern, can be seen as a step towards the design of a deterministic universal construction to execute transaction-based multiprocess programs on top of a multiprocessor. Interestingly, the proposed construction is lock-free (in the sense that it uses no lock).

4.8 Miscellaneous

During the period 2010-2014, I also worked on other topics. I presents here only two of them.

An important concept is the concept of recursivity. I investigated recursivity asynchronous distributed systems where communication is through atomic read/write registers, and any number of processes can commit crash failures [R139,C274]. In such a context and differently from sequential and parallel recursion, the conceptual novelty lies in the fact that the aim of the recursion parameter is to allow each participating process to learn the number of processes that it sees as participating to the task computation.

A second important work is a joint work with Y. Moses [R130]. This work addresses the condition-based simultaneous consensus problem in synchronous message-passing systems (simultaneous means here that all processes have to stop during the very same round). The paper shows that, contrary to what could be hoped, when considering condition-based consensus with simultaneous decision, we can benefit from the best of both actual worlds (either the failure world or the condition world), namely, we cannot benefit from the sum of savings offered by both. Only the best discount applies.

5 Research agenda since 2015: At the frontiers of distributed computing

This part describes research topics in which I am currently involved. In order to be self-contained, this section contains material sections. In this way, they can be read independently.

5.1 Introduction

Preamble Research constitutes one of the main *raisons d'être* of the Universities. It is an obligation for all professors and an absolutely necessary activity to maintain their lectures vivid and up-to-date. I do think that, as a university professor (in informatics), my research activity has to concentrate on understanding computing phenomena, introducing computing-related concepts and clarifying notions that are relevant to computing. That is why I always strove to design algorithms that are as generic and simple as possible. Being generic and simple, they are not bound to specific contexts and consequently capture the essence of the problem they solve (the complementary facet to capture their essence being the determination of the lower bounds associated with the problems they solve). I also do think that genericity and simplicity (that go with beauty and elegance) are first-class citizen criteria when designing solutions to computing problems. I am proud to have some of my algorithms (e.g., message causal ordering, checkpointing, randomized consensus) presented in textbooks written by experts in the domain (e.g., [1, 2, 3, 4]).

[1] Attiya H. and Welch J., Distributed Computing: Fundamentals, Simulations and Advanced Topics, (2d Edition), *Wiley-Interscience*, 414 pages, 2004.
[2] Garg V., Elements of Distributed Computing. *Wiley-Interscience*, 423 pages, 2002.
[3] Cachin Ch., Guerraoui R. and Rodrigues L., Introduction to Reliable and Secure Distributed Programming. *Springer*, 367 pages, 2011.
[4] Ksemkalyani A. and Singhal M., Distributed Computing: Principles, Algorithms, and Systems. *Cambridge University Press*, 738 pages, 2008.

A big picture As already indicated, from a theoretical point of view, the aim of distributed computing is to answer the question: "Which problems can be solved by a set of cooperating entities, and, if the answer is yes, which are the best algorithms for the corresponding problem (best according to some complexity measures)?" Since more than twenty years, this constitutes my research interests. In this context, my research program is focused on fundamental issues, namely investigate the limits of distributed computing in presence of *adversaries* such that asynchrony, failures, anonymity, dynamicity, etc. Lots of results are known for asynchronous systems prone to failures (e.g., the fundamental problem that is consensus has concentrated lots of efforts and its study has provided computer scientists with a deep knowledge on what constitutes a part of the essence of distributed computing). Despite these great advances, lot of work remains to be done. As an example, despite their interest in real applications, dynamicity and anonymity are not yet well understood. My research programmed looks in this direction and focuses on the following domains: distributed computability in asynchronous read/write shared memory systems, distributed computability in asynchronous message-passing systems, relations between the two previous distributed computing models, and concurrent objects. It is of great importance to develop a theory of distributed computing that provides us with concepts and paradigms that help us understand the possibilities and limitations of distributed systems. Such a knowledge is a necessary pre-requisite if one wants to master future (non-trivial) distributed applications.

Today distributed applications are pervasive, some very successful (e.g., Internet, P2P, social networks, cloud computing), and benefit everyone, but the design and the implementation of a lot of them still rely more on "tricks" than on a solid theory. The next generation of distributed applications and services will be more and more complex and requires that we spend today research efforts in establishing sane theoretical foundations to be able to master their design, their properties and their implementation.

One of my leitmotifs is “When something works we have to know why it works, and when it does not work we have to know why it does not work”.

5.2 Distributed computability

Be the communication medium a shared memory of a message-passing system, the aim of distributed computability is to answer the question “what can be computed in a distributed system?” I present below a few fundamental distributed computability problems in which I am interested.

From decision problems to the ranking of sub-consensus tasks I became acquainted with this topic when I tried to establish a connection between the adaptive renaming problem and both the k-set agreement and the (weaker) k-test&set problem [C200,C216]. These works showed that the new space name can be reduced from $2p-1$ (where p is the number of participating processes) to $2p - \lfloor \frac{p}{k} \rfloor$ if we have underlying k-test&set objects, and to $p+k-1$ if we have underlying k-set agreement objects. These results encouraged me to to investigate the relation linking Test&set, adaptive agreement and set agreement. This investigation resulted in [C216]. This “warm-up” research period showed me the richness and the profoundness of the topic. As Test&set, adaptive agreement and set agreement are sub-consensus tasks (i.e., they are weaker than consensus), the “Holy grail” quest is here to establish a hierarchy of sub-consensus tasks (if any), similarly to the hierarchy of consensus tasks as defined by Herlihy. This line of research is fundamental if we want to understand the power of base synchronization primitives. In some sense, we can say that while Herlihy has established the “Mendeleiev’s table” of consensus tasks, the research agenda is here to establish the corresponding table of sub-consensus tasks. The paper that appeared in SIAM JC [R145] is a promising step in this direction, that I continue investigating.

Distributed universal construction A notion of a *universal construction* suited to distributed computing has been introduced by M. Herlihy in his celebrated paper on wait-free synchronization[16]. A universal construction is an algorithm that can be used to wait-free implement any object defined by a sequential specification. Herlihy’s paper shows that the basic system model, which supports only atomic read/write registers, has to be enriched with consensus objects to allow the design of universal constructions. The generalized notion of a *k-universal* construction has been recently introduced by Gafni and Guerraoui[17]. A k-universal construction is an algorithm that can be used to simultaneously implement k objects (instead of just one object), with the guarantee that at least one of the k constructed objects progresses forever. While Herlihy’s universal construction relies on atomic registers and consensus objects, a k-universal construction relies on atomic registers and k-simultaneous consensus objects (which are wait-free equivalent to k-set agreement objects in the read/write system model).

I intend to work on distributed universal constructions, and I already started thinking to build a very general universal construction with the following properties (not satisfied by previous universal constructions). (1) Among the k objects that are constructed, *at least ℓ* objects (and not just one) are guaranteed to progress forever; (2) The progress condition for processes is *wait-freedom*, which means that each correct process executes an infinite number of operations on each object that progresses forever; (3) If any of the k constructed objects stops progressing, all its copies (one at each process) stop in the same state; (4) The proposed construction is *contention-aware*, in the sense that it uses only read/write registers in the absence of contention; (5) It has to be *generous* with respect to the *obstruction-freedom* progress condition, which means that each process is able to complete any one of its pending operations

[16]Herlihy M., Wait-free synchronization. *ACM Transactions on Programming Languages and Systems*, 13(1):124-149, 1991.

[17]Gafni E. and Guerraoui R., Generalizing universality. *Proc. 22nd Int’l Conference on Concurrency Theory (CONCUR’11)*, Springer, LNCS 6901, pp. 17-27, 2011.

on the k objects if all the other processes hold still long enough. Such a construction, that I call (k, ℓ)-universal, should be based on a simple extension of k-simultaneous consensus objects that I (with co-authors) introduced in [R117].

A long lasting open problem: the weakest failure detector for message-passing k-set agreement Assuming each process proposes a value, the k-set agreement requires that each non-faulty process decides a value such that a decided value is a proposed value, and at most k different values are decided. This problem, which generalizes consensus, is impossible to solve in asynchronous crash-prone systems.

While the weakest failure detector for solving the k-set agreement problem in crash-prone asynchronous read/write shared memory systems is known [18], for message-passing systems the weakest failure detectors are known only for the extreme cases $k = 1$ and $k = n - 1$. The important remaining problem is then finding the weakest failure detector for any value of k. I think that answering this question is related to the minimal consistency properties a shared memory has to satisfy in order the k-set agreement problem can be solved (these properties being weaker than the classical properties associated with read/write operations). I started investigating this difficult question, but up to now I have only partial answers [R122,C270,C282]. Hence, despite the efforts of the community, this is an important and lasting consistency problem that belongs to my research program. My hope is that solving this problem will give us a much clearer view of the relation between read/write shared memory systems and message-passing systems.

5.3 Again and again fault-tolerance in the presence of Byzantine processes

The weakest synchrony assumption for Byzantine consensus Let us consider the family of deterministic Byzantine consensus algorithms (i.e., algorithms which are not randomized algorithms). In such a context a fundamental question is the following: "Which are the weakest synchrony assumptions needed to solve consensus in a message-passing system prone to Byzantine process failures?" I solved this important open problem (with Z. Bouzid and A. Mostéfoui) in a PODC paper [C309].

From crash failures to Byzantine failures Borowsky-Gafni's (BG) simulation[19] is a very powerful reduction algorithm designed for asynchronous read/write crash-prone systems, namely, it allows a set of $(t + 1)$ asynchronous sequential processes to wait-free simulate (i.e., despite the crash of up to t of them) an arbitrary number n of processes under the assumption that at most t of them crash. This shows that, in read/write crash-prone systems, t-resilience of decision tasks can be fully characterized in terms of wait-freedom. Said another way, the BG simulation shows that, in read/write systems, a crucial parameter is not the number n of processes, but the upper bound t on the number of processes that may crash in a run.

With D. Imbs, I designed a BG-like simulations in the context of asynchronous *message-passing* systems (which was not addressed before). This, which allowed us to consider crash failures and Byzantine withe same view, was be done in two directions [C327]. The first considers that processes may fail by crashing. Assuming $t < \min(n', n/2)$, the aim is to simulate a system of n' processes where up to t may crash, on top of a basic system of n processes where up to t may crash. The second simulation concerned the case where processes may commit Byzantine failures (up to now the BG simulation considered only process crash failures). Assuming $t < \min(n', n/3)$, the aim is here to simulate a system

[18] Gafni E. and Kuznetsov P., On set consensus number, *Distributed Computing*, 24(3-4):149-163, 2011, and Delporte C., Fauconnier H., Guerraoui R., and Tielmann A., The disagreement power of an adversary. *Distributed Computing* 24(3-4):137-147, 2011.

[19] Borowsky E. and Gafni E., Generalized FLP Impossibility Results for t-Resilient Asynchronous Computations. *Proc. 25th ACM Symposium on Theory of Computing (STOC'93)*, ACM Press, pp. 91-100, 1993, and Borowsky E., Gafni E., Lynch N. and Rajsbaum S., The BG Distributed Simulation Algorithm. *Distributed Computing*, 14:127-146, 2001.

of n' processes where up to t may be Byzantine, on top of a basic system of n processes where up to t may be Byzantine. Moreover, these asynchronous message-passing simulations are direct (in the sense that they do not simulate a shared memory on top of which a suited read/write BG simulation would be used). These constraints are motivated by the fact that they help better understand the deep nature and the difference of crash failures and Byzantine failures in asynchronous message-passing systems.

5.4 Communication abstraction vs agreement abstraction: the two faces of a same coin

It is well-known that consensus (one-set agreement) and total order broadcast are equivalent in asynchronous systems prone to process crash failures. So, I became interested in the following more general question: which is the communication abstraction that "captures" k-set agreement? To this end A. Mostéfaoui, D. Imbs, M Perrin and I introduced a new broadcast communication abstraction, called k-BO-Broadcast, which restricts the disagreement on the local deliveries of the messages that have been broadcast (1-BO-Broadcast boils down to total order broadcast). Hence, in this context, $k = 1$ is not a special number, but only the first integer in an increasing integer sequence [C334]. This establishes a new "correspondence" between distributed agreement problems and communication abstractions, which enriches our understanding of the relations linking fundamental issues of fault-tolerant distributed computing.

We also introduced the "set delivery broadcast" abstraction, which offers the power of atomic registers in crash-prone asynchronous message-passing systems [C335]. This abstraction has a software engineering flavor as it simplifies the message-passing construction of read/write implementable distributed objects.

5.5 Relating theory and practice

A Holy Grail: Is there a "Grand Unified Model"? Synchronous systems and asynchronous systems are the two endpoints of the synchrony axis. With the new adversaries that are anonymity, dynamicity, mobility, etc., finding a distributed computing model that is both realistic and abstract enough (to be tractable) is a real challenge, which will maybe remain an inaccessible scientific "holy grail". Like physicists who are looking for a "Grand Unified Theory" that would encompass all the fundamental concepts of physics, a (much more modest but still) very challenging task would consist in looking for a "Grand Unified Model" for distributed computing. Albeit answering this question seems to much ambitious, it remains at the horizon as a Leibnitz's dream. I started working very recently on this topic ([C290] is a very partial answer to this fundamental issue) and I intend to continue.

Engineering of distributed computing Practitioners and engineers have proposed a number of reusable frameworks and services to implement specific distributed services (from Remote Procedure Calls with Java RMI or SOAP-RPC, to JGroups for group communication, and Apache Zookeeper for primary backup replication). Unfortunately, many of these efforts lack a sound grounding in distributed computation theory (with the notable exceptions of JGroups and Zookeeper), and only provide punctual and partial solutions for a narrow range of services. From my point of view, this is because we still lack a generic framework that is able to unify the large body of fundamental knowledge on distributed computation that has been acquired over the last 20 years. A central issue of distributed computing consists consequently in bridging this gap, by developing a systematic model of distributed computation that organizes the functionalities of a distributed computing system into reusable modular constructs. These constructs should be composable via well-defined mechanisms that maintain sound theoretical guarantees on the resulting system. In relation with my previous research topics, I intend to spend some time and efforts also in this direction, which is crucial for distributed computing engineering. Sound distributed computing engineering is related to the foundations of distributed computing.

Anonymous memory: foundations and algorithms The concept of an anonymous memory was introduced by G. Taubenfed in PODC 2017. An anonymous shared memory M can be seen as an array of atomic registers such that there is no a priori agreement among the processes on the names of the registers. As an example a very same physical register can be known as $M[x]$ by a process p and as $M[y]$ (where $y \neq x$) by another process q. Moreover, the register known as $M[a]$ by a process p and the register known as $M[b]$ by a process q can be the same physical register. It is assumed that each process has a unique identifier that can only be compared for equality.

In such a context, mainly with G. Taubenfeld and D. Imbs, we designed algorithms solving fundamental problems such as mutual exclusion [C345], leader election [C355], and memory de-anonymization [R176]. We established necessary and sufficient conditions for these problems, and in a memory anonymous context. These conditions are primality constraints relating the number of processes ad the size m of the anonymous memory. We also designed a mutual exclusion algorithm in with both the processes and the memory are anonymous [R172].

Concurrent relaxed data structures: the linearizability hierarchy Considering asynchronous shared memory systems in which any number of processes may crash, I recently started working with A. Castaneda and S. Rajsbaum on the identification and the formally definition of relaxations of queues and stacks that can be non-blocking or wait-free while being implemented using only read/write operations. Set-linearizability and interval-linearizability are used to specify the relaxations formally, and precisely identify the subset of executions which preserve the original sequential behavior. They allow for an item to be returned more than once by different operations, but only in case of concurrency, a property property called multiplicity. The stack implementation is wait-free, while the queue implementation is non-blocking. Interval-linearizability is used to describe a queue with multiplicity, with the additional relaxation that a dequeue operation can return *weak-empty*, which means that the queue might be empty. We designed a read/write wait-free interval-linearizable algorithm of for such a concurrent queue. As far as we know, this work is the first that provides formalizations of the notions of multiplicity and weak-emptiness, which can be implemented on top of read/write registers only. This work can be seen as a practical view of the theoretical notions we have introduced in a JACM article [R164].

6 A numerical projection

Miscellaneous	01/2021
h-index Google scholar	61
i-10 index Google scholar	304
Nb citations Google scholar	14141
Articles cited by DBLP	527

Books	12
Chapters in books/encyclopedia	10
Papers in peer-reviewed Journals	179
Papers in peer-reviewed Conf. proceedings	361
Best paper awards (in top conferences)	8
Invited Papers (Int'l Conf + workshops)	34
Supervised PhD Co-supervised PhD	38 10
Invited Courses/Lectures	> 55
PC Member	> 180
PC Chair and Proceedings Editor	> 25
PhD Committees	> 190
Nb of co-authors (as cited by DBLP)	183

Among my 179 articles in journals	01/2021
IEEE Transactions in Parallel and Dist. Systems	16
IEEE Other Trans. (TC, TSE, TKDE, TDCS, TMC)	14
Distributed Computing	9
Journal of Parallel and Distributed Computing	11
Theoretical Computer Science (TCS)	13
Information Processing Letters	18
Parallel Processing Letters	10
JACM, SIAM JC, Algorithmica, JCSS, Springer ToCS Acta Informatica, Information & Computation	18
The Computer Journal	4
Magazines: IEEE Computer, IEEE Software, CACM	6
Articles in French Journals	18
Articles in Asian Journals	3

Among my 361 articles in conferences	01/2021
Published by ACM (not counting short)	39
Published by IEEE	132
Published by Springer LNCS or LIPICS	135
Published by North Holland	11

PODC papers regular, short	18, 20
DISC papers regular, short	20, 5
ICDCS papers	24
SPAA papers	6
SIROCCO papers	15
OPODIS papers	16
SRDS and DSN papers	20

Downloaded chapters of my Springer books		06/2020
Concurrent programming ... algorithms ... and foundations	(published online 12/2012)	46870
Distributed algorithms in message-passing systems	(published online 07/2013)	38221
Fault-tolerant message-passing ... an algorirhmic approach	(published online 09/2018)	

7 Theses

1. *Contribution à l'étude de la coopération entre processus dans les langages et les systèmes informatiques*. Thèse d'État, université de Rennes, 1981.

2. *Conception et réalisation d'une machine-langage de haut niveau adaptée à l'écriture de systèmes*. Thèse de 3ème cycle, université de Rennes, 1975.

8 Guest Editor of Int'l Journals

- Guest editor (with Andrzej Pelc and David Peleg) of the volume 384(2-3):135-286, of the journal *Theoretical Computer Science*, devoted to the publication of selected papers of SIROCCO 2005 (*12th Colloquium on Structural Information and Communication Complexity*.
- Guest editor of a volume of the journal *Theoretical Computer Science* (Volume 561(B):87-144, January 2015) devoted to the publication of selected papers of ICDCN 2013 (*14th Int'l Conference on Distributed Computing and Networking*).
- Guest editor of the Volume 98(8), published in August 2016, of the special issue of the Springer journal *Computing* devoted to selected papers of the international conference NETYS 2014 on networked systems (whose proceedings appeared in Springer LNCS 8593).

9 Books (author)

1. Algorithmique du parallélisme : le problème de l'exclusion mutuelle. *Dunod Ed.*, 160 pages, 1984.
   ▷ English translation: Algorithms for mutual exclusion, *MIT Press,* 107 pages, ISBN 0-262-18119-3, 1986.
   ▷ Spanish translation: Algoritmica del paralelismo, *Omega S.A.*, Barcelona, 1988.
   ▷ Critical review 8704-0237 in *ACM Computing Reviews*, 28(4), 1987.

2. Algorithmes distribués et protocoles. *Eyrolles Ed.*, 142 pages, 1985.
   ▷ English translation: Distributed algorithms and protocols, *Wiley & Sons*, 1988.
   ▷ Critical review in *Software Practice and Experience*, 18(7):711, 1988.

3. Systèmes répartis et réseaux: concepts, outils et algorithmes. *Eyrolles Ed.*, 200 pages, 1987.
   ▷ English translation: Networks and distributed computation, *MIT Press,* 166 pages, 1988.
   ▷ Critical review 8902-0036 in *ACM Computing Reviews*, 30(2):77, 1989.

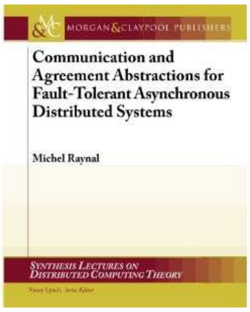
4. Synchronisation et contrôle des systèmes et programmes répartis (with J.-M. Hélary), *Eyrolles Ed.*, 200 pages, 1988.
   ▷ English translation: Synchronization and control of distributed systems and programs, *Wiley & Sons,* 160 pages, 1991.
   ▷ Critical review 9110-0751 in *ACM Computing Reviews*, 32(10):491, 1991.

5. La communication et le temps dans les réseaux et les systèmes répartis. (Tome 1 d'une introduction aux principes des systèmes répartis). *Eyrolles Ed., collection EDF,* 1991, 232 p.
   ▷ Préface de Robert Dautray, membre de l'Académie des Sciences.

6. Synchronisation et état global dans les systèmes répartis. (Tome 2 d'une introduction aux principes des systèmes répartis). *Eyrolles Ed., collection EDF*, 1992, 228 p.

7. Gestion des données réparties : problèmes et protocoles. (Tome 3 d'une introduction aux principes des systèmes répartis). *Eyrolles Ed., collection EDF*, 1993, 200 p.

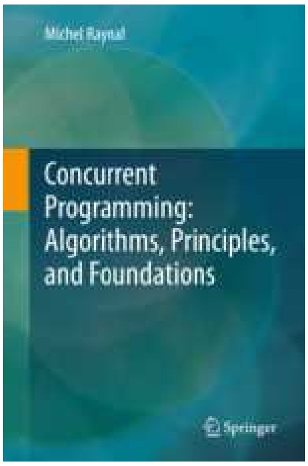
8. Communication and agreement abstractions for fault-tolerant asynchronous distributed systems. *Morgan & Claypool Publishers*, 251 pages, 2010 (ISBN 978-1-60845-293-4).



From the reviews:
As such this book is an ideal textbook for graduate students who have an interest in distributed computing; it will also work as a reference for researchers and interested professionals. The book is very well written and has a rigorous approach to the subject. () This maturity will allow the target audience to truly appreciate the clean and beautiful presentation of this difficult material.
(ACM Computing Reviews, February 18, 2011)

9. Fault-tolerant agreement in synchronous message-passing systems. *Morgan & Claypool Publishers*, 165 pages, 2010 (ISBN 978-1-60845-525-6).

10. Concurrent programming: algorithms, principles and foundations. *Springer*, 515 pages, 2012 (ISBN 978-3-642-32026-2).

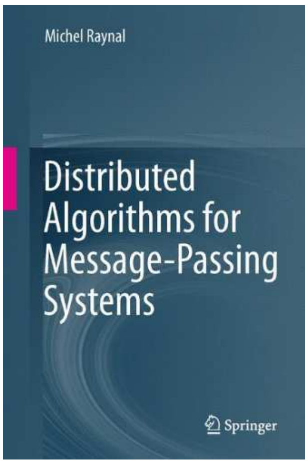


From the reviews:
"Concurrent programming is the study of the methods which will ensure correct interactions. ... Raynal (Univ. of Rennes, France) presents these classical techniques at the beginning of his book, and then moves on to cover such topics as transactional memory and current areas of research like consensus in the face of crash failures. The coverage is very up-to-date, including references through 2010. ... This would be an ideal text for a beginning graduate course. Summing Up: Highly recommended. Graduate students, researchers/faculty, and professionals/practitioners."
(P. Cull, Choice, Vol. 50 (11), August, 2013)

"A very comprehensive treatment of both fundamentals and recent results in concurrent programming is presented in this book. ... The book is well structured, with many examples to help the reader. Each chapter starts with a short presentation of the content and a list of keywords, and concludes with a summary of the main points and results. ... I can recommend this book ..."
(Sergei Gorlatch, ACM Computing Reviews, June, 2013)

Since its online publication on December 23, 2012, until June 2020, there have been a total of 46870 chapter downloads for the eBook version of this book on SpringerLink.

11. Distributed algorithms for message-passing systems. *Springer*, 510 pages, 2013 (ISBN: 978-3-642-38122-5).

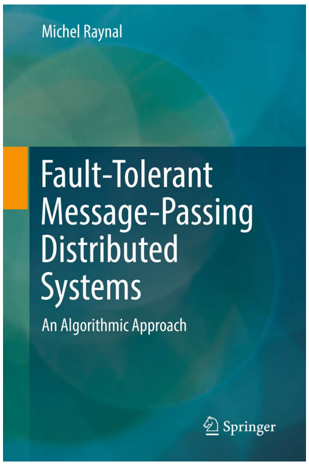


From the reviews:
"The book presents in well structured manner the basic concepts and algorithms currently used in distributed systems based on message passing. ... The book can be used as textbook by undergraduate students in distributed systems. What distinguishes this book from similar ones are the text accessibility and the well organization of a classical material. Many figures and pseudo-codes are helping the understanding of the algorithms."
(Dana Petcu, zbMATH, Vol. 1282, 2014)

This book offers balanced coverage of the major topics encountered in courses on modern distributed computations. The text is primarily intended for courses on distributed systems; it can be used for both undergraduate and postgraduate courses. Overall, this title is an instructive and valuable book that deserves to be studied.
(Dimitrios Katsaros, ACM Computing Reviews, June, 2014)

Since its online publication on July 13, 2013 until June 2020, there have been a total of 38221 chapter downloads for the eBook version of this book on SpringerLink.

12. Fault-tolerant message-passing distributed systems: an algorithmic approach. *Springer*, 492 pages, 2018 (ISBN: 978-3-319-94140-0).



From the reviews:

Although my entire career has centered on building sophisticated distributed systems, I've never been particularly good at explaining complex protocols. Yet this need is unavoidable in my graduate courses at Cornell I have in mind topics like proving bounds on Byzantine Agreement under various models, or showing how self-stabilizing token rings repair themselves after damage. The authors treatment here is a breath of fresh air: a principled exploration of the important concepts and protocols, intelligently organized. I plan to draw on this book when preparing my lectures, and to use it as a resource for students who want to explore the ideas in greater depth. The many case studies are valuable too, because they show how the protocols and theory can be applied to real-world systems, and how they can sometimes be specialized to leverage features of the hardware for speedups. The author has created a wonderful resource, and I highly recommend it.

Ken Birman, Dept. of Computer Science, Cornell University.

This brilliant book is a must-read for all students and researchers in the area of distributed systems. The author has presented fundamental issues in designing fault-tolerant distributed systems in a clear and concise manner. The book provides intuition behind complex ideas as well as rigorous proofs for mathematically minded readers. The book is also quite timely due to the emergence of cloud computing and blockchains. I would recommend this book for practitioners as well as theoreticians in distributed computing.

Vijay K. Garg, The University of Texas at Austin.

10 Chapters in books and articles in encyclopdia

1. Chapter "Consensus in Asynchronous Distributed Systems: A Concise Guided tour", in *Advances Distributed Systems*, Springer, LNCS 1752, pp. 33-47, 2000 (with R. Guerraoui, M. Hurfin, A. Mostéfaoui, R. Oliveira and A. Schiper).
2. Chapter "Time in Distributed Systems: Models and Algorithms" in *Advances Distributed Systems*, Springer, LNCS 1752, pp. 33-47, 2000 (with Paulo Verissimo -Lisbon University-).
3. Chapter "Reliable Logical Clocks for Unreliable Process Groups" in "Dependable Network Computing" pp. 93-108, *Kluwer Academic Press*, 2001 (with A. Mostéfaoui, M. Takizawa -Tokyo Denki University-).
4. Chapitre "Une introduction à l'algorithmique distribuée des systèmes asynchrones", *Encyclopédie Vuibert* de l'informatique et des systèmes d'information, pp. 179-194, 2006 (ISBN 978-2-7117-4846-4).
5. Article "Failure Detectors for Asynchronous Distributed Systems: an Introduction". *Wiley Encyclopdia of Computer Science and Engineering*, Vol. 2, pp. 1181-1191, 2009 (ISBN 978-0-471-38393-2).
6. Article " Set agreement", Second Edition of Encyclopedia of Algorithms, pp. 1956-1959, *Springer*, 2016 (ISBN 978-1-4939-2863-7).
7. Article " Distributed Snapshots", Second Edition of Encyclopedia of Algorithms, pp. 581-586, *Springer*, 2016 (ISBN 978-1-4939-2863-7).
8. Article "Messages adversaries", Second Edition of Encyclopedia of Algorithms, pp. 1272-1276, *Springer*, 2016 (ISBN 978-1-4939-2863-7).
9. Chapter "A Short Visit to Distributed Computing Where Simplicity is Considered as a First Class Property", in *The French School of Programming*, B. Meyer Editor, Springer (2021).

11 Articles in journals

Among my 179 publications in journals, 17 are in French and 162 in english. **The title of the papers I consider as the most important are in bold characters**. *Moreover, except for extremely few, the author list in my journal and conference papers obeys alphabetical order.*

[R1] Une expression de la synchronisation pour les types abstraits. *RAIRO Revue Bleue/Computer Science*, 12(4):307-316, 1978.

[R2] An experience in implementing abstract data types. *Software Practice and Experience*, 11:315-320, 1980 (with M. Banatre, A. Couvert, c D. Herman).

[R3] Types in a mixed language system. *BIT*, (now *Nordic Journal of Computing*) 23(2):246-256, 1981 (with Ph. Darondeau, P. Le Guernic).

[R4] Une analyse de la spécification de la coopération entre processus par variables partagées. *Techniques et Science Informatiques (TSI)*, 1(3):201-210, 1982.

[R5] **Structured specification of communicating systems.** *IEEE Transactions on Computers*, vol.C32(2):120-133, 1983 (with G. von Bochmann –Université de Montréal–).

[R6] Un algorithme d'exclusion mutuelle pour une structure logique en anneau. *Techniques et Science Informatiques (TSI)*, 4(5):471-474, 1985.

[R7] A distributed algorithm to prevent mutual drift between n logical clocks. *Information Processing Letters*, 24:199-202, 1987.

[R8] Parcours et apprentissage dans un réseau de processus communicants. *Techniques et Science Informatiques (TSI)*, 5(2):127-140, 1987 (with J.-M. Hélary, A. Maddi, N. Plouzeau).

[R9] Producteur-consommateur : quelques solutions réparties. *Techniques et Science Informatiques (TSI)*, 6(3):231-241, 1987 (with N. Plouzeau, J.-P. Verjus).

[R10] Calcul distribué d'un extremum et du routage associé dans un réseau quelconque. *Rairo Informatique Théorique et Applications*, 21(3):1-22, 1987 (with J.-M. Hélary, A. Maddi).

[R11] A distributed algorithm for mutual exclusion in an arbitrary network. *The Computer Journal*, 31(4):289-295, 1988 (with J.M. Hélary and N. Plouzeau).

[R12] **Un schéma abstrait d'itération répartie, application au calcul des chemins de valeurs minimales.** *Techniques et Science Informatiques (TSI)*, 8(3):259-268, 1989 (with J.-M. Hélary).

[R13] Prime numbers as a tool to design distributed algorithms. *Information Processing Letters*, 33(1):53-58, 1989.

[R14] Simulation répartie : schémas d'exécution pour un modèle à processus. *Techniques et Science Informatiques (TSI)*, 9(5):383-398, 1990 (with Ph. Ingels).

[R15] Vers la construction raisonnée d'algorithmes répartis, le cas de la terminaison. *Techniques et Science Informatiques (TSI)*, 10(3):203-209, 1991 (with J.-M. Hélary).

[R16] La communication causale dans les systèmes répartis, protocoles fondés sur le comptage. *Revue Réseaux et Informatique Répartie*, 1(1):87-99, 1991.

[R17] **The causal ordering abstraction and a simple way to implement it.** *Information Processing Letters*, 39:343-351, 1991 (with A. Schiper –EPFL, Lausanne–, S. Toueg –Cornell University–).

[R18] A debugging tool for distributed Estelle programs. *Journal of Computer Communications*, 16(5):328-333, 1993 (with M. Hurfin, N. Plouzeau).

[R19] Protocoles simples pour l'implémentation répartie des sémaphores. *Annales des Télécommunications*, 48(5-6):260-273, 1993.

[R20] Un noyau réparti pour les applications fondées sur la progression d'un temps virtuel. *Revue Réseaux et Informatique Répartie*, 3(2):145-168, 1993 (with Ph. Ingels, C. Maziero).

[R21] Towards the construction of distributed detection programs with an application to distributed termination. *Distributed Computing*, 7(3):137-147, 1994 (with J.-M. Hélary).

[R22] **A general scheme for token and tree based distributed mutual exclusion algorithms.** *IEEE Transactions on Parallel and Distributed Systems*, 5(11):1185-1196, 1994 (with J.-M. Hélary, A. Mostéfaoui).

[R23] Déterminer un état global dans un système réparti. *Annales des télécommunications*, 49(7-8):460-469, (with J.-M. Hélary, A. Mostéfaoui).

[R24] An implementation of global flush primitives using counters. *Parallel Processing Letters*, 5(2):171-178, 1995 (with M. Ahuja –Univerity of California, San Diego–).

[R25] A graph-based characterization of communication modes in distributed executions. *Journal of Foundations of Computing and Decision Sciences*, 20(1):3-20, 1995 (with R. Baldoni -University of Roma-).

[R26] **Specification and verification of dynamic properties in distributed computations**. *Journal of Parallel and Distributed Computing,* 28(2): 173-185, 1995 (with O. Babaoğlu -University of Bologna, Italy-).

[R27] Distributed algorithms for static and dynamic termination detection. *Bulletin of the Polish Academy of Sciences*, 43(3):363-380, 1995 (with J. Brzezinsky -University of Poznan-, J.-M. Hélary).

[R28] **On-the-fly analysis of distributed computations**. *Information Processing Letters*, 54:267-274, 1995 (with E. Fromentin, C. Jard, G.-V., Jourdan).

[R29] Semantics of recovery lines for backward recovery in distributed systems. *Annales des Télécommunications*, 50(10-11):874-887, 1995 (with J. Brzezinsky -University of Poznan-, J.-M. Hélary).

[R30] **Deadlock models and a general algorithm for distributed deadlock detection**.
Journal of Parallel and Distributed Computing, 31(2):112-125, 1995. (with J. Brzezinsky -University of Poznan-, J.-M. Hélary, M. Singhal –Ohio State University, Colombus–).

[R31] An introduction to snapshot algorithms in distributed computing, *Journal of Distributed Systems Engineering*, 2(4):224-233, 1995 (with A. Kshemkalyani, M. Singhal -Ohio state university-).

[R32] Causal delivery of messages with real-time data in unreliable networks, *Real-Time Systems Journal*, 10(3): 245-262, 1996 (with R. Baldoni –La Sapienza, Roma–, A. Mostéfaoui).

[R33] Logical time: capturing causality in distributed systems. *IEEE Computer*, 29(2):49-57, 1996 (with M. Singhal –Ohio State University, Colombus–).

[R34] An optimistic protocol for a linearizable distributed shared memory system. *Parallel Processing Letters*, 6(2):65-278, 1996 (with M. Mizuno, M. Neilsen -Kansas state University-).

[R35] A unified framework for expressing and detecting run-time properties of distributed computations. *Journal of Systems Software*, 33(3):287-298, 1996 (with O. Babaoglu –Univerità di Bologna, Italy–, E. Fromentin).

[R36] **From group communication to transactions in distributed systems**. *Communications of the ACM*, 39(4):84-90, 1996 (with A. Schiper –EPFL, Lausanne–).

[R37] On-the-fly detection of a class of words-based patterns in labeled dags. *Applied Mathematics and Computer Science*, 7(1):205-216, 1997 (with M. Hurfin).

[R38] Adaptive checkpointing in message-passing distributed systems. *Int'l Journal of Systems Science*, 28(11):1145-1161, 1997. (with R. Baldoni, –University of Roma–, J.-M. Hélary, A. Mostéfaoui).

[R39] k-Arbiter: a safe and general scheme for h out of k mutual exclusion. *Theoretical Computer Science*, 193(1-2):97-112, 1998. (with R. Baldoni –La Sapienza, University of Roma–, Y. Manabe –NTT, Japan–).
▷ Critical review in *ACM Computing Reviews*, 39(11):572.

[R40] An adaptive causal ordering algorithm suited to mobile computing environments. *Journal of Parallel and Distributed Computing*,41(1):190-204, 1997 (with M. Singhal, R. Prakash –Ohio State University, Colombus–).

[R41] **Shared global states in distributed computations**. *Journal of Computer and System Sciences*, 55(3):522-528, 1997 (with E. Fromentin).

[R42] Revisiting the non-blocking atomic commitment problem in distributed data management systems. *Ingénierie des Systèmes d'Information*, 5(6):639-659, 1997.

[R43] A suite of definition for consistency criteria in distributed shared memories. *Annales des Télécommunications*, 52(11-12):652-661, 1997 (with A. Schiper –EPFL, Lausanne–).

[R44] Efficient delta-causal broadcasting. *Journal of Computer Systems Science and Engineering*, vol. 13(5):263-270, 1998 (with R. Baldoni, R. Prakash, M. Singhal –Ohio State University, Colombus–).

[R45] Consistent records in asynchronous computations. *Acta Informatica*, 35:441-455, 1998 (with R. Baldoni, J.-M. Hélary).

[R46] De la validation atomique au consensus: une approche synthétique. *Techniques et Science Informatiques (TSI)*, 17(3):279-298, 1998 (with R Guerraoui, A. Schiper –EPFL Lausanne–).

[R47] Points de contrôle dans les systèmes répartis : concepts et protocoles. *Techniques et Science Informatiques (TSI)*, 17(10):1223-1246, 1998 (with A. Mostéfaoui, J.-M. Hélary).

[R48] **Efficient distributed detection of conjunction of local predicates**. *IEEE Transactions on Software Engineering*, 24(8):664-677, 1998. (with M. Hurfin, M. Mizuno, M. Singhal).
▷ Critical review in *ACM Computing Reviews*, 39(10):527.

[R49] **Consistency issues in distributed checkpoints.** *IEEE Transactions on Software Engineering*, 25(4):274-281, 1999 (with R. Netzer –Brown University–, J.-M. Hélary).

[R50] Non-blocking atomic commitment in distributed systems: a tutorial based on a generic protocol. *Journal of Computer Systems Science and Engineering*, 15(2):77-86, 2000.

[R51] **Normality: a consistency criterion for concurrent objects.** *Parallel Processing Letters*, 9(1):123-134, 1999 (with V. Garg – University of Texas, Austin–).

[R52] An adaptive architecture for causally consistent distributed services. *Distributed Systems Engineering Journal*, 6(2):63-70, 1999 (with M. Ahamad, G. Thia-Kime).

[R53] **Communication-induced determination of consistent snapshots**. *IEEE Transactions on Parallel and Distributed Systems*, 10(9):865-877, 1999 (with J.-M. Hélary, A. Mostéfaoui).

[R54] A simple and fast asynchronous consensus protocol based on a weak failure detector. *Distributed Computing*, 12(4):209-223, 1999 (with M. Hurfin).

[R55] Group membership failure detection: a simple protocol and its probabilistic analysis. *Distributed Systems Engineering Journal*, 6(3):95-102, 1999 (with F. Tronel).

[R56] Restricted failure detectors: definition and reduction protocols. *Information Processing Letters*, 72:91-97, 1999 (with F. Tronel).

[R57] **Communication-based prevention of useless checkpoints in distributed computations**. *Distributed Computing*, 13(1):29-43, 2000. (withJ.-M. Hélary, R. Netzer, A. Mostéfaoui).

[R58] **From binary consensus to multivalued consensus in asynchronous message-passing systems**. *Information Processing Letters*, 73:207-213, 2000. (with A. Mostéfaoui, F. Tronel).

[R59] **Computing global functions in asynchronous distributed systems with perfect failure detectors**. *IEEE Transactions on Parallel and Distributed Systems*, 11(9):897-909, 2000 (with J.-M. Hélary, M. Hurfin, A. Mostéfaoui, F. Tronel).

[R60] Fault-tolerant consensus-based total order multicast. *IEEE Transactions on Parallel and Distributed Systems*. 13(2):147-157, 2001 (with U. Fridzke, Ph. Ingels, A. Mostéfaoui).

[R61] Impossibility of scalar clock-based communication-induced checkpointing protocols ensuring the RDT property. *Information Processing Letters*, 80:105-111, 2001 (with R. Baldoni, A. Mostéfaoui, J.-M. Hélary).

[R62] Rollback-dependency trackability: a minimal characterization and its protocol. *Information and Computation*, 165(2):144-173, 2001 (with R. Baldoni, J.-M. Hélary).

[R63] Direct dependency-based determination of consistent global checkpoints. *Journal of Computer Systems Science and Engineering*, 16(1):43-50, 2001 (with R. Baldoni, G. Cioffi –La Sapienza, University of Roma–, J.M. Hélary).

[R64] The logically instantaneous communication mode: a communication abstraction. *Future Generation of Computer Systems*, 17(6):669-678, March 2001. (with A. Mostéfaoui, P. Verissimo).

[R65] Mastering agreement problems in distributed systems. *IEEE Software*, 18(4):40-47, 2001 (with M. Singhal, Ohio State university).

[R66] Consistent checkpointing for transaction systems. *The Computer Journal*, 44(2):92-100, 2001 (with R. Baldoni, F. Quaglia –La Sapienza, University of Roma–).

[R67] **Leader-based consensus**. *Parallel Processing Letters*, 11(1):95-107, 2001 (with A. Mostéfaoui).

[R68] Fundamentals of distributed computing: a practical tour of vector-clock systems. *IEEE Distributed Systems Online* (http://www.computer.org/dsonline), 3(2):1-18, 2002 (with R. Baldoni).

[R69] **A versatile family of consensus protocols based on Chandra-Toueg's unreliable failure detectors**. *IEEE Transactions on Computers*, 51(4):395-408, 2002. (with M. Hurfin, A. Mostéfaoui).

[R70] **Interval consistency of asynchronous distributed computations**. *Journal of Computer and System Sciences,* 64(2):329-349, 2002 (with A. Mostéfaoui, J.-M. Hélary).

[R71] Real-time based consistency models for distributed objects. *Journal of Computer Systems Science and Engineering*, 17(2):133-142, 2002 (with M. Ahamad, F. Torres, Georgia Tech).

[R72] An introduction to oracles for asynchronous distributed systems. *Future Generation Computer Systems*, 18(6):757-767, 2002 (with E. Mourgaya, A. Mostéfaoui).

[R73] A Note on the determination of the immediate predecessors in a distributed computation. *Int'l Journal of Foundations of Computer Science (IJFCS)*, 13(6):865-972, 2002 (with E. Anceaume, J.M. Hélary).

[R74] Token-based sequential consistency. *Journal of Computer Systems Science and Engineering*, 17(6):359-365, 2002.

[R75] **Efficient causality-tracking timestamping**. *IEEE Transactions on Knowledge and Data Engineering*, 15(5): 1239-1250, 2003 (with R. Baldoni, J.-M. Hélary, G. Melideo –La Sapienza, University of Roma–).

[R76] Early stopping in global data computation. *IEEE Transactions on Parallel and Distributed Systems*, 14(9):909-921, 2003 (with C. Delporte-Gallet, H. Fauconnier –LIAFA, Paris VII–, J.-M. Hélary).

[R77] Atomic broadcast in asynchronous crash-recovery distributed systems and its use in quorum-based replication. *IEEE Transactions on Knowledge and Data Engineering*, 15(5):1206-1217, 2003 (with L. Rodrigues, Lisboa, Portugal).

[R78] **Conditions on input vectors for consensus solvability in asynchronous distributed systems**. *Journal of the ACM*, 50(6):922-954, November 2003. (with A. Mostéfaoui, S. Rajsbaum).

[R79] Consensus in Byzantine asynchronous systems. *Journal of Discrete Algorithms*, 1(2):185-210, 2003 (with R. Baldoni, J.-M. Hélary, L. Tanguy).

[R80] **Condition-based consensus solvability: a hierarchy of conditions and efficient protocols**. *Distributed Computing*, 17(1):1-20, 2004 (with A. Mostéfaoui, S. Rajsbaum, M. Roy).

[R81] Necessary and sufficient condition for transforming limited accuracy failure detectors. *Journal of Computer and System Sciences,* 68:123-133, 2004 (with E. Anceaume, A. Fernandez, A. Mostéfaoui, G. Neiger).

[R82] The information structure of indulgent consensus. *IEEE Transactions on Computers*, 53(4):453-466, 2004 (with R. Guerraoui).

[R83] A weakest failure detector-based asynchronous consensus protocol for $f < n$. *Information Processing Letters*, 90(1):39-46, 2004 (with R. Friedman, A. Mostéfaoui).

[R84] Quorum-based multi-invocation model for replicated objects. *Concurrent Engineering: Research and Application Journal*, June 2004. (with T. Enokido, K. Hori, M. Takizawa –Tokyo Denki Univeriity–).

[R85] Wait-free computing: an introductory lecture. *Future Generation Computer Systems*, 21(5):655-663, 2005.

[R86] Asynchronous bounded lifetime failure detectors. *Information Processing Letters*, 94(2):85-91, 2005 (with R. Friedman –The Trechnion–, A. Mostéfaoui).

[R87] A short introduction to failure detectors for asynchronous distributed systems. *ACM Sigact News, Distributed Computing Column*, 36(1):53-70, 2005.

[R88] Stabilizing mobile philosophers. *Information Processing Letters*, 95(1):299-306, 2005 (with A. Datta –University of Nevada, Las Vega–, M. Gradinariu).

[R89] $\Diamond\mathcal{P}_{mute}$-Based consensus for asynchronous Byzantine systems. *Parallel Processing Letters*, 15(1-2):162-182, 2005 (with R. Friedman, A. Mostéfaoui).

[R90] **Simple and efficient oracle-based consensus protocols for asynchronous Byzantine systems**. *IEEE Transactions on Dependable and Secure Computing*, 2(1):46-56, 2005 (with R. Friedman, A. Mostéfaoui).

[R91] **Synchronous condition-based consensus**. *Distributed Computing*, 18(5):325-343, 2006 (with A. Mostéfaoui, S. Rajsbaum).

[R92] Time-free and timer-based assumptions can be combined to get eventual leadership. *IEEE Transactions on Parallel and Distributed Systems*, 17(7):656-666, 2006 (with A. Mostéfaoui, C. Travers).

[R93] **A Time-free assumption to implement eventual leadership**. *Parallel Processing letters*, 16(2):189-208, 2006 (with A. Mostéfaoui, E. Mourgaya, C. Travers).

[R94] From $\Diamond\mathcal{W}$ to Ω: a simple bounded quiescent reliable broadcast-based transformation. *Journal of Parallel and Distributed Computing*, 67(1):125-129, 2007 (with A. Mostéfaoui, S. Rajsbaum, C. Travers).

[R95] **The Alpha of indulgent consensus**. *The Computer Journal*, 50(1):53-67, 2007 (with R. Guerraoui).

[R96] An adaptive programming model for fault-tolerant distributed computing. *IEEE Transactions on Dependable and Secure Computing*, 4(1):18-31, 2007 (with S. Gorender, R. Macedo, Salvador de Bahia, Brazil).

[R97] On the respective power of $\Diamond\mathcal{P}$ and $\Diamond\mathcal{S}$ to solve one-shot agreement problems. *IEEE Transactions on Parallel and Distributed Systems*, 18(5):589-597, 2007 (with R. Friedman, A. Mostéfaoui).

[R98] **Distributed agreement problems and their connection with error-correcting codes**. *IEEE Transactions on Computers*, 56(7):865-875, 2007 (with Friedman R., A. Mostéfaoui, S. Rajsbaum).

[R99] Design and performance evaluation of efficient consensus protocols for mobile Ad Hoc networks. *IEEE Transactions on Computers*, 56(8):1055-1070, 2007 (with Jiannong Cao, J. Yang, W. Wu -Hong-Kong-).

[R100] An impossibility about failure detectors in the iterated immediate snapshot model. *Information Processing Letters*, 108(3):160-164, 2008 (with S. Rajsbaum, C. Travers).

[R101] Modularity: a first class concept to address distributed systems. *ACM Sigact News*, 39(2):91-110, 2008. (with A.-M. Kermarrec, R. Friedman).

[R102] Using asynchrony and zero-degradation to speed up indulgent consensus protocols. *Journal of Parallel and Distributed Computing*. 68(7):984-996, 2008 (with J. Cao, W. Wu, J. Lin, Hong-Kong Polytechnic University).

[R103] On the computability power and the robustness of set agreement-oriented failure detector classes. *Distributed Computing*, 21(3):201-222, 2008 (with A. Mostéfaoui, S. Rajsbaum, C. Travers).

[R104] Anonymous graph exploration without collision by mobile robots. *Information Processing Letters*. 109:98-103, 2008 (with R. Baldoni, F. Bonnet, A. Milani).

[R105] **The combined power of conditions and information on failures to solve asynchronous set agreement**. *SIAM Journal of Computing*, 38(4):1574-1601, 2008 (with A. Mostéfaoui, S. Rajsbaum, C. Travers).

[R106] From adaptive renaming to set agreement. *Theoretical Computer Science*, 410(14-15): 1328-1335, 2009 (with E. Gafni, A. Mostéfaoui, C. Travers).

[R107] On the fly estimation of the processes that are alive in an asynchronous message-passing system. *IEEE Transactions on Parallel and Distributed Systems*, 20(6):778-787, 2009 (with A. Mostéfaoui, G. Trédan).

[R108] **Revisiting simultaneous consensus with crash failures.** *Journal of Parallel and Distributed Computing*, 69:400-409, 2009 (with Y. Moses, the Technion, Israel).

[R109] A note on atomicity: boosting Test&Set to solve Consensus. *Information Processing Letters*, 109(12):589-591, 2009 (with D. Imbs).

[R110] **Conditions for set agreement with an application to synchronous systems**. *Springer Journal Computer Science and Technology*, 24(3):418-433, 2009 (with F. Bonnet).

[R111] Eventual clusterer: a modular approach to designing hierarchical consensus protocols in MANETs. *IEEE Transactions on Parallel and Distributed Systems*, 20(6):753-765, 2009 (with J. Cao, W. Wu, Hong-Kong Polytechnic University).

[R112] **Strongly terminating early-stopping k-set agreement in synchronous systems with general omission failures.** *Theory of Computing Systems*, 17(1):259-287, 2010 (with Ph. Raipin Parvedy, C. Travers).

[R113] **A timing assumption and two t-resilient protocols for implementing an eventual leader service in asynchronous shared-memory systems**, *Algorithmica*, 56(4):550-576, 2010 (with A. Fernandez, E. Jimenez, G. Trédan).

[R114] **Narrowing power vs efficiency in synchronous set agreement: relationship, algorithms and lower bound**. *Theoretical Computer Science*, 411(1):58-69, 2010. (with Mostéfaoui, C. Travers).

[R115] A methodological construction of an efficient sequentially consistent distributed shared memory. *The Computer Journal*, 53(9):1523-1534, 2010. (with V. Cholvi, A. Fernandez, E. Jimenez, and P. Manzano, Madrid).

[R116] From an asynchronous intermittent rotating star to an eventual leader. *IEEE Transactions on Parallel and Distributed Systems*, 21(9):1290-1303, 2010. (with A. Fernandez, Madrid).

[R117] **The k-simultaneous consensus problem**. *Distributed Computing*, 22(3):185-196, 2010. (with Y. Afek, E. Gafni, S. Rajsbaum, C. Travers).

[R118] **A simple proof of the necessity of the failure detector Σ to implement an atomic register in asynchronous message-passing systems**. *Information Processing Letters*, 110(4):153-157, 2010, (with F. Bonnet).

[R119] Eventual leader election with weak assumptions on initial knowledge, communication reliability and synchrony. *Springer Journal of Computer Science and Technology (JCST)*, 25(6):1267-1281, 2010 (With A. Fernandez and E. Jimenez, Madrid).

[R120] Software transactional memories: an approach for multicore programming. *Springer Journal of Supercomputing*, 57(2):203-215, 2011. (With D. Imbs).

[R121] **The Price of anonymity: optimal consensus despite asynchrony, crash and anonymity**. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 6(4):Article 23 (28 pages), October 2011. (With F. Bonnet).

[R122] On the road to the weakest failure retector for k-set agreement in message-passing systems. *Theoretical Computer Science*, 412(33):4272-4284, 2011. (With Bonnet).

[R123] Failure detectors to solve asynchronous k-set agreement: a glimpse of recent results. *Bulletin of the European Association of Theoretical Computer Science (EATCS)*, 103(1):75-95, 2011.

[R124] A liveness condition for concurrent objects: x-wait-freedom. *Concurrency and Computation: Practice and experience*, 23:2154-2166, 2011. (With D. Imbs).

[R125] The renaming problem in shared memory systems: an introduction. *Elsevier Computer Science Review*, 5:229-251, 2011. (With A. Castaneda and S. Rajsbaum, UNAM Mexico).

[R126] Implementing a regular register in an eventually synchronous distributed system prone to continuous churn. *IEEE Transactions on Parallel and Distributed Systems*, 23(1):102-109, 2012. (With S. Bonomi, R.Baldoni, Roma).

[R127] **Hélp when needed, but no more: an efficient partial snapshot algorithm**. *Journal of Parallel and Distributed Computing*, 72(1):1-12, 2012. (With D. Imbs.)

[R128] **Virtual world consistency: a condition for STM systems (with a versatile protocol with invisible read operations)**. *Theoretical Computer Science*, 444:113-127, 2012. (with D. Imbs.)

[R129] Specifying and implementing an eventual leader service for dynamic systems. *International Journal of Web and Grid Services*, 8(3):204-224, 2012. (With Larrea M., Soraluze I. and Cortiñas R., San Sebastian, Spain.)

[R130] **No double discount: condition-based simultaneity yields limited gain**. *Information and Computation*, 214:45-56, May 2012. (With Y. Moses –The Techion, Israël).

[R131] From the *Happend-Before* relation to the causal ordered set abstraction. *Journal of Parallel and Distributed Computing*, 72:791-795, 2012. (With S. Pomarez and J.R. Perez Cruz, Puebla University, Mexico).

[R132] Computability in distributed computing: a tutorial. *ACM Sigact News, Distributed Computing Column*, 43(3), pp. 80-102, 2012. (With M. Herlihy –Brown University–, S. Rajsbaum –UNAM, Mexico–)

[R133] **Anonymous asynchronous systems: the case of failure detectors**, *Distributed Computing*, 26(3):141-158, 2013. (With F. Bonnet).

[R134] **The weakest failure detector to implement a register in asynchronous systems with hybrid communication**. *Theoretical Computer Science*, 512:130-142, 2013. (With D. Imbs).

[R135] Towards a universal construction for transaction-based multiprocess programs. *Theoretical Computer Science*, 496:154-169, 2013. (With T. Crain, D. Imbs).

[R136] Trust-aware peer sampling: performance and privacy tradeoffs. *Theoretical Computer Science*, 512:67-83, 2013. (With Frey D., Jegou A., Kermarrec A.-M., and Stainer J.)

[R137] **Power and limits of distributed computing shared memory models.** *Theoretical Computer Science*, 509:3-24, 2013. (With M. Herlihy, S. Rajsbaum).

[R138] Computabilidad distribuida. (In Spanish.) *Miscelánea Mathematica*, 56:113-144, 2013 (ISSN-1665-5478). (With M. Herlihy –Brown University–, S. Rajsbaum –UNAM, Mexico–) .

[R139] An introductory tutorial to concurrency-related distributed recursion. *Bulletin of the European Association of Theoretical Computer Science (EATCS)*, 111:57-75, (2013). (With S. Rajsbaum –UNAM, Mexico–).

[R140] **Signature-free asynchronous binary Byzantine consensus with $t < n/3$, $O(n^2)$ messages, and $O(1)$ expected time.** *Journal of ACM*, 62(4), Article 31, 21 pages, 2015. (With A. Mostéfaoui and H. Moumen).

[R141] Failure detectors in homonymous distributed systems (with an application to consensus). *Journal of Parallel and Distributed Computing*. 83:83-95, 2015. (With S. Arevalo, A. Fernãndez, D. Imbs, and E. Jimenez–Madrid–).

[R142] Predicate detection in asynchronous distributed systems: a probabilistic approach. *IEEE Transactions on Computers*, 65(1):173-186, 2016. (With J. Cao -HK Polytechnic-, W. Zhu -Wuhan university-, China). DOI:10.1109/TC.2015.2409839.

[R143] **Intrusion-tolerant broadcast and agreement abstractions in the presence of Byzantine processes**. *IEEE Transactions on Parallel and Distributed Systems*, 27(4):1085-1098 (2016) (With A. Mostéfaoui).

[R144] Distributed slicing in dynamic systems. *IEEE Transactions on Parallel and Distributed Systems*. 27(4):1030-1043 (2016) (With A. Fernández, V. Gramoli, E. Jiménez, and A.-M. Kermarrec).

[R145] **Generalized symmetry breaking, tasks, and non-determinism in concurrent objects**. *SIAM Journal of Computing*, 45(2):379-414 (2016) (With A. Castañeda, D. Imbs, and S. Rajsbaum).

[R146] Implementing set objects in dynamic distributed systems. *Journal of Computer and System Sciences*, 82(5):654-689, 2016. (With R. Baldoni, S. Bonomi, Università La Sapienza, Roma).

[R147] Read/Write shared memory abstraction on top of an asynchronous Byzantine message-passing system. *Journal of Parallel and Distributed Computing*, 93-94:1-9 (2016) (With D. Imbs, S. Rajsbaum and J. Stainer).

[R148] A fast contention-friendly binary search tree. *Parallel Processing Letters*, 26(3):1-17 (2016). (With T. Crain, V. Gramoli, Sydney University, Australia).

[R149] **Distributed universality**. *Algorithmica*. 76(2):502-535, 2016, (With J. Stainer and G. Taubenfeld). DOI 10.1007/s00453-015-0053-3.

[R150] A necessary condition for Byzantine k-set agreement. *Information Processing Letters*, 116(12):757-759, 2016. (With Z. Bouzid and D. Imbs.)

[R151] **Trading t-resilience for efficiency in asynchronous Byzantine reliable broadcast**. *Parallel Processing Letters*, Vol. 26(4), 8 pages (2016) (With D. Imbs)

[R152] A distributed leader election algorithm in crash-recovery and omissive systems. *Information Processing Letters*, 118:100-104, 2017. (With C. Fernández-Campusano, M. Larrea, and R. Cortiñas, University of the Basque Country).

[R153] Distributed universal constructions: a guided tour. *Bulletin of the European Association of Theoretical Computer Science (EATCS)*, 121(1):64-96 (2017)

[R154] **Atomic read/write memory in signature-free Byzantine asynchronous message-passing systems**. In *Theory of Computing Systems*, Springer, 60(4):677-694 (2017) (With A. Mostéfaoui, M. Petrolia, and Cl. Jard, Université de Nantes). DOI: 10.1007/s00224-016-9699-8.

[R155] **Signature-free asynchronous Byzantine systems: from multivalued to binary consensus with $t < n/3$, $O(n^2)$ messages, and constant time**. *Acta Informatica*, 54(5):501-520 (2017) DOI 10.1007/s00236-016-0269-y (With A. Mostéfaoui.)

[R156] **From wait-free to arbitrary concurrent solo executions in colorless distributed computing**. *Theoretical Computer Science*, 683:1-21 (2017) (With M. Herlihy –Brown University, USA–, S. Rajsbaum –UNAM, Mexico–, and J. Stainer).

[R157] Agent-based broadcast protocols for wireless heterogeneous node networks. *Computer Communications*, 115:51-63 (2018) (With A. Bouabdallah, J. Bourgeois, H. Lakhlef.)

[R158] Energy-efficient composite event detection in wireless sensor networks. *IEEE Communications Letters*, 22(1):177-180 (2018) (With Zu Weiping -Wuhan University China, and Cao J., Polytechnic University Hong Kong).

[R159] Randomized k-set agreement in crash-prone and Byzantine asynchronous systems. *Theoretical Computer Science*, 709:81-98 (2018) (With A. Mostéfaoui and H. Moumen).

[R160] **Anonymous obstruction-free (n, k)-set agreement with $(n - k + 1)$ atomic read/write registers**. *Distributed Computing*, 31(2):99-117 (2018) (With Z. Bouzid and P. Sutra.)

[R161] Implementing snapshot objects on top of crash-prone asynchronous message-passing systems. *IEEE Transactions on Parallel and Distributed Systems*, 29(9):2033-2045 (2018) (With C. Delporte and H. Fauconnier –IRIF, Paris 7–, and S. Rajsbaum, UNAM, Mexico.)

[R162] A simple object that spans the whole consensus hierarchy. *Parallel Processing Letters* 28(2), 1850006, 9 pages (June 2018). (With A. Mostéfaoui and M. Perrin.)

[R163] Extending causal consistency to any object defined by a sequential specification. *Bulletin of the European Association of Theoretical Computer Science (EATCS)*, 125:81-92 (2018) (With A. Mostéfaoui and M. Perrin).

[R164] **Unifying concurrent objects and distributed tasks: Interval-linearizability**. *Journal of the ACM*, 65(6), Article 45, 42 pages (2018) (With A.Castañeda and S. Rajsbaum, UNAM, Mexico.)

[R165] A speculation-friendly binary search tree. *Concurrency Practice and Experience*, 31(4), 20 pages (2019) (With Crain T. and Gramoli V., University of Sydney, Australia)

[R166] Time-efficient read/write register in crash-prone asynchronous message-passing systems. *Computing*, 10(1):3-17 (2019). (With A. Mostéfaoui and M. Roy.)

[R167] Making local algorithms wait-free: the case of ring coloring. *Theory of Computing Systems*, Springer, 63(2):344-365 (2019). (With A. Castañeda, C. Delporte and H. Fauconnier, Université Paris 7, and S. Rajsbaum, UNAM, Mexico.)

[R168] **Vertex coloring with communication constraints in synchronous broadcast networks.** *IEEE Transactions on Parallel and Distributed Systems*, 30(7):1672-1686 (2019). (With Lakhlef H. and Taïani F.)

[R169] Crash-tolerant causal broadcast in $O(n)$ messages. *Information Processing Letters*. Vol. 151, 105837, 6 pages (2019). (With A. Mostéfaoui and M. Perrin, Université de Nantes, France, and Jiannong Cao, Hong Kong Polytechnic University.)

[R170] An Informal visit to the wonderful land of consensus numbers and beyond. *Bulletin of the European Association of Theoretical Computer Science (EATCS)*, 129:168-192 (2019)

[R171] **Mastering concurrent computing through sequential thinking: A half-century evolution.** *Communications of the ACM*, Vol. 63(1):78-87 (2020) (With S. Rajsbaum, UNAM, Mexico.)

[R172] **Mutual exclusion in fully anonymous shared memory systems**. *Information Processing Letters*, Vol. 158, 105938, 7 pages (2020) (With Gadi Taubenfeld–Interdisciplinary Center, Herliyan Israël–).

[R173] Collisions are preferred: RFID-based stocktaking with a high missing rate. *IEEE Transactions on Mobile Computing*, 19(7):1544-1554 (2020) (With Zhu Weiping, Meng Xing, Peng Xiaolei, Cao Jiannong, Wuhan University –China– Hong Kong PolyU).

[R174] 60 Years of mastering concurrent computing through sequential thinking. *ACM Sigact News*, Vol. 51(2), pp. 59-88 (2020) (With S. Rajsbaum, UNAM, Mexico.)

[R175] An eventually perfect failure detector for networks of arbitrary topology connected with ADD channels using time-to-live values. *Parallel Processing Letters*, 30(2), 2050006, 23 pages (With K. Vargas and S. Rajsbaum, UNAM, Mexico.)

[R176] **Leader-based de-anonymization of an anonymous read/write memory**. *Theoretical Computer Science*, 836(10):110-123 (2020). (With E. Godard and D. Imbs, LIS, Université d'Aix- Marseille, and G. Taubenfeld, Interdisciplinary Center, Herzliya, Israel).

[R177] **Money transfer made simple.** *Bulletin of the European Association of Theoretical Computer Science (EATCS)*, Vol. 132, pp. 22-43 (October 2020) (With A. Auvolat, D. Frey, and F. Taïani.)

[R178] On the versatility of Bracha's Byzantine reliable broadcast algorithm. *Parallel Processing Letters*, To appear (2021)

[R179] **Set-constrained delivery broadcast: definition, abstraction power, and computability limits**. *Theoretical Computer Science* To appear (2021) (With D. Imbs, A. Mostéfaoui, M. Perrin.)

——— Articles currently submitted ———

[R180] **Contention-related crash failures: definitions, agreement algorithms, and impossibility results**. Submitted to *Theoretical Computer Science* (With Anaïs Durand and Gadi Taubenfeld).

[R181] Byzantine-tolerant causal broadcast. Submitted to *Theoretical Computer Science* (With A. Auvolat, D. Frey, and F. Taïani.)

[R182] On the weakest information on failures to solve mutual exclusion and consensus in asynchronous crash/prone read/write systems. Submitted to *Journal of Parallel and Distributed Computing*. (With C. Delporte and H. Fauconnier, Université Paris Sorbonne.)

[R183] **k-Immediate snapshot and x-set agreement: how are they related?** Selected for the special issue of *Information & Computation* devoted to the conference SSS 2020. (With C. Delporte and H. Fauconnier –Université Paris–, and S. Rajsbaum –UNAM, Mexico–.)

[R184] Shared-memory mutual exclusion: a short history in six dates. Submitted to *Elsevier Computer Science Review*. (With G. Taubenfeld, Interdisciplinary Center, Herzliya, Israël).

[R184] **Concurrent queues and stacks from read/write operations.** Submitted to *Distributed Computing*, (2021) (With A.Castañeda and S. Rajsbaum, UNAM, Mexico.)

12 Articles in conferences

A ratio a/n appearing after a title means that a papers were accepted from a total of n submissions. As for journals, titles of the **papers I consider as the most important are in bold characters**.

[C1] Minimizing command language in a system based on a high level language. *IFIP Conference on Command Languages*, *North-Holland 1975*, pp. 117-135, Lund (Sweden), 1974 (with D. Herman, L. Trilling, J.-P. Verjus).

[C2] A standard prelude for a multi-user operating system based on Algol 68. *Int'l Conference on Algol 68,*n pp. 190-202, Stillwater (Oklahoma) 1975 (with Y. Bekkers, D. Herman, J.-P. Verjus).

[C3] Communication between processes and the process management in SAR. *Conference on Experience with Algol 68*, Liverpool (U.K.) 1975 (with Y. Bekkers, D. Herman, J.-P. Verjus).

[C4] Faults and remedies of compile-time protection methods. *5th Annual IFIP W.G.2.1 Conference on the Implementation and the Design of Algorithmic Languages*, pp. 102-110, Guidel (France), 1977 (with Y. Bekkers, Ph. Darondeau).

[C5] A model of inter-context control transfer. *7th Congress of the Gesellschatft für Informatik*, Springer, 1977 (with Ph. Darondeau, T. Muntean).

[C6] Types abstraits et pluralité de leurs représentations à l'exécution. *3ème Colloque International sur la programmation*, Paris, *Dunod*, pp. 263-278, 1978 (with M. Banatre, A. Couvert, D. Herman).

[C7] Présentation et évaluation du projet SOC : Un système d'objets conservés typés. *Actes du Congrès AFCET*, Gif-sur-Yvette, *Ed. Hommes et Techniques*, pp. 446-456, 1978 (with M. Banatre, A. Couvert, D. Herman).

[C8] Abstract specification of communication systems. *Proc. 1st IEEE Int'l Coference on Distributed Computing Systems*, Huntsville, (Alabama) pp. IEEEE Computer Press, pp. 339-346, 1979 (with Ph. Darondeau, P. Le Guernic).

[C9] Synchronisation and protection features for data abstraction. *Proc. 10th Congress Gesellschatft für Informatik*, Springer, pp. 277-291, Saarbrucken (RFA), 1980 (with D. Herman).

[C10] Éléments d'un langage adapté à la communication entre processus. *Actes du congrès AFCET-Informatique*, Nancy, *Ed. Hommes et Techniques)*, pp.667-676, 1980 (with P. Le Guernic).

[C11] Un exercice dans la conception d'un programme en termes de processus coopérants. *Actes du congrès AFCET-Informatique*, pp. 117-126, Gif sur Yvette, 1981.

[C12] **Detection of stable properties in distributed applications.** (25/128). *Proc. 6th ACM Sigact-Sigops Symposium on Principles of Distributed Computing (PODC'87)*, ACM Press, pp. 125-136, 1987 (with J.-M. Hélary, C. Jard, N. Plouzeau).

[C13] An algorithm to detect token loss on a logicial ring and to regenerate lost tokens. *Int'l Conference on Parallel Processing and Applications*, l'Aquila (Italy), North-Holland, *Parallel Processing and Applications*, pp. 457-467, 1987 (with G. Rubino).

[C14] The rudiments of object distribution in a distributed system. Invited Talk. *2nd Int'l Conference on Computer and Information Sciences*, Istanbul, *Nova Science Pub.*, New York, 1988 (with C. Jard).

[C15] General and efficient decentralized consensus protocols. *2nd Int'l Workshop on Distributed Algorithms (WDAG)*, Springer LNCS 312, pp. 41-57, 1987 (with J.-C. Bermond, J.-C. Konig).

[C16] Controlling information transfers in distributed applications, application to deadlock detection. *Int'l Conference on Parallel Processing and Applications*, l'Aquila (Italy), North-Holland, *Parallel Processing and Applications*, pp. 85-92, 1987 (with J.-M. Hélary A. Maddi).

[C17] Depth-first traversal and virtual ring construction in distributed parallel systems. *Proc. IFIP WG 10.3 Conference on Parallel Processing*, North-Holland, *Parallel Processing*, pp. 333-346, Pisa (Italy), 1988 (with J.-M. Hélary).

[C18] Les parcours répartis de réseaux : un outil pour la conception de protocoles. *CFIP 88, (Conference Francophone sur l'Ingénierie des protocoles)*, Bordeaux, *Eyrolles*, pp. 159-170, 1988 (with J.-M. Hélary).

[C19] Assigning distinct identities to sites of an anonymous distributed system. *Proc. IEEE Conference on Future of Distributed Computing Systems (FTDCS)*, IEEE Computer Press, pp. 82-86, Hong-Kong, 1988 (with J.-M. Hélary).

[C20] The meaning of synchronous distributed algorithms run on asynchronous distributed systems. *Proc. 3rd Int'l Symposium on Computers and Information Sciences*, (1988), Izmir (Turkey) *Nova Science Pub. Ed.*, New York, pp. 307-316, 1989 (with M. Adam, Ph. Ingels).

[C21] Experimentation on a parallel machine is helpful to analyse distributed algorithms. *Proc. Int'l Workshop on Parallel and Distributed Algorithms*, Bonas (France), North-Holland, pp. 243-250, 1989 (with M. Adam, Ph. Ingels, Cl. Jard, J.-M. Jézéquel).

[C22] Distributed algorithms: their nature and the problems encountered. *Proc. Int'l Workshop on Parallel and Distributed Algorithms*, Bonas (France), North-Holland, pp. 179-185, 1989.

[C23] Characterization of a particular class of distributed snapshots. *Proc. Int'l Conference on Computing and Information*, North-Holland *Computing and Information*, Toronto (Canada), 1989 (with J.-M. Hélary, N. Plouzeau).

[C24] Vers une problématique de l'algorithmique répartie, Invited Talk. *Colloque Franco-Brésilien sur les Systèmes Répartis*, Florianopolis (Brazil), 1989.

[C25] Distribution of a global invariant in a parallel distributed system. *Parallel Computing 89*, Leiden (Holland), North-Holland, pp. 597-602, 1989.

[C26] Distributed computers in the design and analysis of distributed algorithms. *1st European Workshop on Hypercube and Distributed Computers*, Rennes, (France), *North-Holland*, 1989.

[C27] Computing particular snapshots in distributed systems. (110/354).
Proc. 9th Int'l IEEE Phœnix Conference on Computers and Communications, IEEE Computer press, pp. 116-123, Scottsdale (Arizona) 1990 (with J.-M. Hélary, N. Plouzeau).

[C28] Order notions and atomic multicast in distributed systems. *Proc. 2d IEEE Int'l Conference on future trends of Distributed Computing Systems (FTDCS)*, IEEE Computer Press pp. 420-427, Cairo (Egypt), 1990.

[C29] Implementation and evaluation of distributed synchronization on distributed memory parallel machines. (48/102). *2nd European Conference on Distributed Memory Computing EDMCC2*, Springer LNCS 487, pp. 304-314, Munich (Germany), 1991 (with A. Couvert, R. Pédrono).

[C30] **A distributed solution to the k out of m resources allocation problem.** (60/166). *Proc. Int'l Conference on Computing and Information (ICCI'91)*, Springer LNCS 497, pp. 509-522, Ottawa (Canada), 1991.

[C31] Implementing semaphores on a distributed memory parallel machine. *Proc. Parallel Computing 91*, North-Holland, pp. 407-412, London (UK), 1991.

[C32] Implementation of a distributed debugger for Estelle programs. *Proc. of the ERCIM Workshop on Distributed Systems*, pp. 47-51, Lisboa (Portugal), 1991 (with M. Hurfin, N. Plouzeau).

[C33] **A general method to define quorums.** (85/301). *Proc. 12th IEEE Int'l Conference on Distributed Computing Systems (ICDCS'92)*, IEEE Computer Press, pp. 657-664, Yokohama (Japan), 1992 (with M. Mizuno, M. Neilsen).

[C34] Synchronization and concurrency measures for distributed computations. (85/301). *Proc. 12th IEEE Int'l Conference on Distributed Computing Systems (ICDCS'92)*, IEEE Computer Press, pp. 700-709, Yokohama (Japan), 1992 (with M. Mizuno, M. Neilsen).

[C35] EREBUS: a debugger for asynchronous distributed computations. *Proc. 3rd IEEE Conference on future trends in Distributed Computing Systems (FTDCS'92)*, IEEE Computer Press, pp. 93-98, Taipei (Taiwan), 1992 (with M. Hurfin, N. Plouzeau).

[C36] FLORIA: a distributed kernel for virtual time driven applications. *4th Int'l Conference on Computing and Information (ICCI'92)*, IEEE Computer Press, pp. 457-460, Toronto (Canada), 1992 (with Ph. Ingels, C. Mazeiro).

[C37] Distributed evaluation: a tool for constructing distributed programs. *Int'l Symposium on Theory of Computing and Systems*, Springer LNCS 601, pp. 184-194, (Rodeh, Galil, Dolev Eds), Haifa (Israel), 1992 (with J.-M. Hélary).

[C38] Discrete event simulation on distributed memory parallel machines. *Proc. Workshop on Advances in Parallel Algorithms*, IOS Press, Sofia, pp. 176-183, 1992 (with C. Maziero).

[C39] From distributed mutual exclusion to readers and writers algorithms. *Proc. 7th Int'l Conference ISCIS*, Antalya (Turkey), pp. 129-136, 1992.

[C40] Que peut apporter l'algorithmique répartie lorsque l'on est confronté aux défaillances. Invited Talk. *Congrès AFCET 93*, Tome *Sécurité et Sûreté Informatiques*, pp.69-78, Versailles (France), 1993.

[C41] Concepts et problèmes de l'algorithmique répartie. Invited Talk. *Congrès AFCET 93*, Tome *Fondements du parallélisme*, pp.1-15, Versailles (France), 1993.

[C42] **Termination detection in a very general distributed computing model.** (63/299). *13th IEEE Int'l Conference on Distributed Computing Systems (ICDCS)*, IEEE Computer Press, pp. 374-382, Pittsburgh, 1993 (with J. Brzezinsky -Poznan university-, J.-M. Hélary).

[C43] **Detecting atomic sequences of predicates in distributed computations.** (16/57). *Proc 3rd ACM Conference on Parallel and Distributed. Debugging*, ACM SIGPLAN Notices 28(12):33-42, San Diego (CA), 1993 (with M. Hurfin, N. Plouzeau).

[C44] **How to find his way in the jungle of consistency criteria for distributed object memories (or how to escape from Minos' labyrinth)**. *Proc. 4th IEEE Conference on Future Trends of Distributed Computing Systems (FTDCS)*, IEEE Computer Press, pp. 340-346, Lisboa (Portugal), 1993. (with M. Mizuno).

[C45] Causal multicast in overlapping groups: towards a low cost approach. *Proc. 4th IEEE Conference on Future Trends of Distributed Computing Systems (FTDCS)*, IEEE Computer Press, pp. 136-142, Lisboa (Portugal), 1993. (with A. Mostéfaoui).

[C46] **Specification and verification of behavioral aptterns in distributed computations.** (21/95). *Proc. 4th Int'l IFIP WG 10.4 Conference on Dependable Computing for Critical Applications (DCCA 4)*, Springer series on Dependable Computing and Fault-Tolerant Systems, Vol. 9, pp.271-290, San Diego (CA), 1994 (with O. babaoğlu - University of Bologna-).

[C47] Definition and implementation of a flexible communication primitives for distributed programming. (32/79). *Proc. Int'l IFIP WG 10.3 Conference on Applications in Parallel and Distributed Computing*, North-Holland, pp. 115-124, Caracas (Venezuela), 1994 (with A. Mostéfaoui).

[C48] An Efficient implementation of sequentially consistent distributed shared memories. (32/79). *Proc. Int'l IFIP WG 10.3 Conference on Applications in Parallel and Distributed Computing*, North-Holland, pp. 145-154, Caracas (Venezuela), 1994 (with M. Mizuno, G. Singh, M. Neilsen -University of Kansas-).

[C49] On granularity of events in distributed computations. Invited Talk. *Proc. Workshop on Specification of Parallel programs*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science (K.M. Chandy and G. Blelloch Eds), 18:115-124, Princeton, 1994 (with M. Hurfin, N. Plouzeau).

[C50] La mise au point des programmes répartis: un problème, quelques solutions. Invited Talk. *7mes Journées Internationales des Sciences Informatiques)*, pp. 283-302, Tunis, 1994.

[C51] On-the-fly replay: a practical paradigm and its implementation for distributed debugging. (59/197). *Proc. 6th IEEE Int'l Symposium on Parallel and Distributed Processing (SPDP)*, IEEE Computer Press, System Track, pp. 266-272, Dallas (TX), 1994 (with M. Hrfin, N. Plouzeau, O. Gerstel, S. Zaks -Technion-).

[C52] **Inevitable global states: a concept to detect unstable properties of distributed computations**. (59/197). *6th IEEE Int'l Symposium on Parallel and Distributed Processing (SPDP)*, IEEE Computer Press, Theory Track, pp. 242-248, Dallas (TX), 1994 (with E. Fromentin).

[C53] On the fly testing of regular patterns in distributed computations. (28/118). *Proc. 23th Int'l Conference on Parallel Processing (ICPP)*, Vol.II, pp. 73-78, Pennsylvania State University, Illinois, 1994 (with E. Fromentin, V. Garg, A. Tomlinson -Austin-).

[C54] Sequential consistency in distributed systems. Invited Talk. *Int'l Workshop on Theory and Practice in DIStributed Systems*, Springer LNCS 938, pp. 224-243, Dagstuhl (Germany), 1994 (with M.Mizuno, J. Zhou –Kansas State University).

[C55] Analyzing and debugging distributed executions. Invited Talk. *Proc. Int'l Conference Parallel and Distributed Computing*, KU Press, pp. 13-25, Koweit City, 1995.

[C56] **Characterizing and detecting the set of global states seen by all the observers of a distributed computation**. (60/170). *Proc. 15th Int'l IEEE Conference on Distributed Computing. Systems*, IEEE Computer Press, pp. 431-438, Vancouver (Canada), 1995 (with E. Fromentin).

[C57] A look at replaying techniques for distributed executions. Invited Talk. *Proc. 2d Int'l Workshop on Automated and Algorithmic Debugging*, Saint-Malo (France), 1995 (with N. Plouzeau).

[C58] Properties of communication modes in distributed systems. (74/165). *Proc. IEEE Conference on Algorithms and Architectures for Parallel Processing (ICA3P2)*, IEEE Compute Press, pp. 501-505, Brisbane (Australia), 1995 (with R.Baldoni).

[C59] An introduction to the analysis and debug of distributed computations. (74/165). *Proc. IEEE Conference on Algorithms and Architectures for Parallel Processing (ICA3P2)*, IEEE Compute Press, pp. 545-554, Brisbane (Australia), 1995 (with E. Fromentin, N. Plouzeau).

[C60] A general definition of deadlocks for distributed systems. (74/165). *Proc. ACM-IEEE Conference on Algorithms and Architectures for Parallel Processing (ICA3P2)*, IEEE Compute Press, pp. 324-332, Brisbane (Australia), 1995 (with J. Brzezinski -Poznan university-, J.-M. Hélary).

[C61] Recognition of patterns on control flows for debugging distributed executions. *Proc. 2d Int'l Workshop on Automated and Algorithmic Debugging* (AADEBUG), Saint-Malo (France) 1995 (with E. Fromentin, C. Jard, G.V. Jourdan).

[C62] Debugging distributed executions using language recognition. (20/90). *Proc. 24th Int'l Conference on Parallel Processing (ICPP), Pennsylvania State University*, Vol. II pp.55-62 (Software Track), Oconomowoc (WI), 1995 (with O. Babaoglu -University of Bologna-, E. Fromentin).

[C63] Efficient causally ordered communications for multimedia real-time applications. (20/68). *Proc. 4th IEEE Int'l Symposium on High Performance Distributed Computing (HPDC)*, IEEE Computer Press, pp.140-147, Washington DC, 1995 (with R. Baldoni, A. Mostéfaoui).

[C64] Deadlocks in distributed systems: request models and definitions. Invited Talk. *Proc. 5th IEEE Int'l Conference on Future Trends of Distributed Computing Systems (FTDCS)*, IEEE Computer Press, pp. 186-193, Cheju (South Korea), 1995 (with J. Brzezinski -Poznan university-, J.-M. Hélary).

[C65] Modeling consistent checkpoints and the domino effect in large scale distributed systems. *Proc. 5th IEEE Int'l Conference on Future Trends of Distributed Computing Systems (FTDCS)*, IEEE Computer Press, pp. 314-323, Cheju (South Korea), 1995 (with R. Baldoni, J. Brzezinski -Poznan university-, J.-M. Hélary, A. Mostéfaoui).

[C66] **Expressing and detecting control flows properties of distributed computations**. (99/249). *Proc. 7th Int'l IEEE Symposium on Parallel and Distributed Processing (SPDP)*, IEEE Computer Press, pp. 432-438, San Antonio, (TX), 1995 (with E. Fromentin, V. Garg).

[C67] **From causal consistency to sequential consistency in shared memory systems.** *Proc. 15th Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS)*, Springer LNCS 1026, pp. 180-194, Bangalore (India), 1995 (with A. Schiper, EPFL).

[C68] A synchronization measure for communication modes in distributed systems. *Proc. 14th Int'l Conference on Applied Informatics*, pp. 204-207, Innsbruck (Austria), 1996 (with R. Baldoni, G. Cioffi –La Sapienza, University of Roma–).

[C69] Causal multicast in unreliable networks for multimedia applications. *15th IEEE Int'l Phoenix Conference on Computers and Communications*, IEEE Computer Press, pp. 51-57, Phoenix (AZ) 1996 (with R. Baldoni–La Sapienza, University of Roma–, A. Mostéfaoui).

[C70] An informal introduction to distributed algorithms. Invited Talk. *Int'l Conference on Distributed systems*, Salvador de Bahia (Brazil), 1996.

[C71] **An efficient causal ordering algorithm for mobile computing environment**. (86/295). *Proc. 15th IEEE Int'l Conference on Distributed Computing Systems (ICDCS)*, IEEE Computer Press, pp. 744-751, Hong-Kong, 1996 (with M. Singhal, R. Prakash –Ohio state university, Comombus–).

[C72] Quorum-based solutions to the h out of k mutual exclusion problem. *4th Int'l Workshop on Parallel Systems and Algorithms (PASA4)*, World Scientific Pub., Lubeck (Germany) 1996 (with R. Baldoni, Y. Manabe -ntt, Japan-).

[C73] Broadcast with time and causality constraints for multimedia applications. *22d EUROMICRO Conference*, IEEE Computer Press, pp.617-624, Pragua (Czech Republic), 1996 (with R. Baldoni, M. Singhal, R. Prakash –Ohio state Sniversity, Colombus–).

[C74] Weakening transactions to fit the need of cooperative applications. *Int'l Conference on Information, Systems, Analysis and Synthesis (ISAS)*, pp.531-538, Orlando (FL), 1996 (with G. Thia-kime).

[C75] Detecting diamonds necklaces in labeled directed acyclic graphs: a problem from distributed debugging. *22th Int'l Workshop on Graph Theoretic Concepts in Computer Science*. Springer LNCS 1197, pp. 211-223, Lago di Como (Italy), 1996 (with M. Hurfin).

[C76] Reducing the number of messages logged in uncoordinated checkpointing algorithms. (26/76). *2d European Dependable Computing Conference (EDCC2)*, Springer LNCS 1150, pp.353-364, Taormina (Italy), 1996 (with A. Mostéfaoui).

[C77] Efficient distributed detection of conjunction of local predicates in asynchronous computations. (81/217). *Proc. 8th Int'l IEEE Symposium on Parallel and Distributed Processing (SPDP)*, IEEE Computer Press, pp. 588-594, New-Orleans, October 1996 (with M. Hurfin, M. Mizuno, M. Singhal).

[C78] A suite of formal definitions for consistency criteria in distributed shared memories. *Proc. 9-th Int'l IEEE Conference on Parallel and Distributed Computing Systems (PDCS)*, IEEE Computer Press, pp. 125-131, Dijon (France), 1996 (with A. Schiper).

[C79] Static and dynamic adaptation of transactional consistency. (70/180). *Proc. 30th International Conference on Systems Sciences (HICSS)*, Vol I, pp. 533-542, Maui (Hawai), 1997 (with O. Theel).

[C80] Consistent state restoration in shared memory systems. (59/167). *Proc. Int'l IEEE Conference on Advances in Parallel and Distributed Computing (APDC)*, IEEE Computer Press, pp. 330-337, Shanghai (China), 1997 (with R. Baldoni, A. Mostéfaoui).

[C81] Real-time dependable decisions in timed asynchronous distributed systems. *Proc. 3rd Int'l IEEE Workshop on Object-Oriented Real-time Distributed Systems (WORDS)*, IEEE Computer Press, pp. 283-290, Newport Beach (CA) 1997.

[C82] Revisiting the non-blocking atomic commitment problem in distributed systems. Invited Talk. *Proc. 2d IPPS IEEE Workshop on Fault-Tolerant Parallel and Distributed Systems*, IEEE Computer Press, pp. 116-133, Genève (Switzerland), 1997.

[C83] **A communication-induced checkpointing protocol that ensures the rollback-dependency trackability property**. (42/122). *27th IEEE Symposium on Fault-Tolerant Computing Systems (FTCS)*, IEEE Computer Press, pp. 68-77, Seattle (WA), 1997 (with R. Baldoni, J.-M. Hélary, A. Mostéfaoui).

[C84] From serializable to causal transactions for collaborative applications. *23th EUROMICRO Conference*, IEEE Computer Press, pp. 314-321, Budapest (Hungary), 1997 (with G. Thiakime, M. Ahamad -Georgia Tech-).

[C85] **Preventing useless checkpoints in distributed computations**.(20/72). *16th IEEE Symposium on Reliable Distributed Systems (SRDS)*, IEEE Computer Press, pp. 183-190, Durham (VG), 1997 (with J.-M. Hélary, A. Mostéfaoui, R. Netzer).

[C86] A case study of agreement problems in distributed systems. *Proc. 2d Int'l IEEE Workshop on High Assurance Systems Engineering (HASE)*, IEEE Computer Press, pp. 156-164, Washington DC, 1997.

[C87] **Virtual precedence in asynchronous systems: concept and applications**. (20/59). *11th Int'l Workshop on Distributed Algorithms (WDAG)*, Springer LNCS 1320, pp. 170-184, Saarbrucken (Germany), 1997 (with J.-M. Hélary, A. Mostéfaoui).

[C88] Implementing train applications based on the distributed shared memory paradigm. *Proc. 14th IFIP/IFAC Int'l Conference on Distributed Computer Control Systems (DCCS)*, pp. 191-196, Seoul (South Korea), 1997 (with G. Bae, G. Thiakime).

[C89] Cycle prevention in distributed checkpointing. Invited Talk. *Proc. Int'l symposium On Principles Of DIstributed Systems (OPODIS)*, Hermès pp. 309-318, Chantilly (France), 1997, (with J.-M. Hélary, A. Mostéfaoui).

[C90] Exploiting write semantics in implementing partially replicated causal objects. (28/89). *Proc. 6th EUROMICRO Conference on Parallel and Distributed Processing (PDP'98)*, IEEE Computer Press, pp. 157-163, Madrid (Spain), 1998.

[C91] A practical building block for solving agreement problems in asynchronous distributed systems. *Proc. 17th IEEE Int'l Performance, Computing and Communications Conference*, IEEE Computer Press, pp. 25-31, Phoenix (AZ), 1998 (with M. Hurfin, F. Tronel).

[C92] Asynchronous protocols to meet real-time constraints: is it really sensible? How to proceed? Invited Talk. *Proc. First Int'l IEEE Symposium on Object-Oriented Real-Time Distributed Computing (ISORC)*, IEEE Computer Press, pp. 290-297, Kyoto (Japan), 1998 (with M. Hurfin).

[C93] An adaptive protocol for implementing causally consistent distributed services. (68/305). *18th IEEE Int'l Conference on DIStributed Computing Systems (ICDCS)*, IEEE Computer Press, pp. 86-93, Amsterdam (Netherlands), 1998 (with M. Ahamad, G. Thiakime).

[C94] **Communication-induced determination of consistent snapshots**. (47/199). *28th IEEE Int'l Symposium on Fault-Tolerant Computing (FTCS*, IEEE Computer Press, pp. 208-217, Munich (Germany), 1998 (with J.-M. Hélary, A. Mostéfaoui).

[C95] **Fault-tolerant total order multicast for asynchronous groups**. *17th IEEE Symposium on Reliable Distributed Systems, (SRDS)*, IEEE Computer Press, pp. 228-234, Purdue, 1998 (with U. Fridzke, Ph. Ingels, A. Mostéfaoui).

[C96] Consensus in asynchronous systems where processes can crash and recover. *17th IEEE Symposium on Reliable Distributed Systems, (SRDS)*, IEEE Computer Press, pp. 280-286, Purdue, 1998 (with M. Hurfin, A. Mostéfaoui).

[C97] Lifetime-based consistency for distributed objects. (28/87). *12th Int'l Symposium on DIStributed Computing (DISC), (Previously WDAG)*, Springer LNCS 1499, pp. 378-392, Andros (Greece), 1998 (with M. Ahamad, F. Torres -Georgia Tech-).

[C98] Probabilistic analysis of a group failure detection protocol. *Proc. 4th Int'l IEEE Workshop on Object-Oriented Real-time Distributed Systems (WORDS 99)*, IEEE Computer Press, pp. 156-162, Santa Barbara (CA), 1999 (with R. Bollo, J.-P. Le Narzul, F. Tronel).

[C99] Rollback dependency trackability: visible characterizations. (27/95). *18th ACM SIGACT-SIGOPS Int'l Symposium on Principles of Distributed Computing (PODC'99)*, ACM Press, pp. 33-42, Atlanta, 1999 (with R. Baldoni, J.-M. Hélary).

[C100] **Timed consistency for shared distributed objects**. (27/95). *18th ACM SIGACT-SIGOPS Int'l Symposium on Principles of Distributed Computing (PODC'99)*, ACM Press, pp. 163-172, Atlanta, 1999 (with M. Ahamad, F. Torres -Georgia Tech-).

[C101] **On classes of problems in asynchronous distributed systems with process crashes**. Best Paper Award (57/173). *19th IEEE Int'l Conference on Distributed Computing Systems (ICDCS)*, IEEE Computer Press, pp. 470-477, Austin (TX), 1999 (with E. Fromentin, F. Tronel).

[C102] Logical clocks for unreliable process groups. *Int'l Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)*, pp. 2272-2278, Las Vegas (NV), 1999 (with A. Mostéfaoui).

[C103] Efficient logically instantaneous communications in asynchronous distributed systems. (47/103). *5th Int'l Conference on Parallel Computing Technologies (PaCT'99)*, Springer LNCS 1662, pp. 258-270, St-Petersburg (Russia), 1999 (with A. Mostéfaoui, P. Verissimo).

[C104] Consistent Lamport's clocks for asynchronous groups with process crashes. (47/103). *5th Int'l Conference on Parallel Computing Technologies (PaCT'99)*, Springer LNCS 1662, pp. 98-107, St-Petersburg (Russia), 1999 (with A. Mostéfaoui, M. Takizawa -Tokyo-).

[C105] Illustrating the use of vector clocks in property detection: an example and a counter-example. (115(*regular papers*)/341). *Proc. 5th Int'l European Parallel Computing Conference (EUROPAR'99)*, Springer LNCS 1685, pp. 806-814, Toulouse (France), 1999.

[C106] Consistent checkpointing for distributed databases. (115 (*regular papers*)/341). *Proc. 5th Int'l European Parallel Computing Conference (EUROPAR'99)*, Springer LNCS 1685, pp. 450-458, Toulouse (France), 1999 (with R. Baldoni, F. Quaglia -University of Roma-).

[C107] A general framework to solve agreement problems. (26/61). *18th IEEE Symposium on Reliable Distributed Systems, (SRDS'99)*, IEEE Computer Press, pp. 56-67, Lausanne (Switzerland), 1999 (with M. Hurfin, R. Macedo, F. Tronel).

[C108] Simple vector clocks are limited to solve some causality-related problems. Invited talk. *Proc. Int'l Symposium On Principles Of DIstributed Systems (OPODIS'99)*, pp. 199-208, Hanoi, Vietnam (1999)

[C109] Direct dependency-based determination of consistent global checkpoints. *Proc. Int'l Symposium On Principles Of DIstributed Systems (OPODIS'99)*, pp. 11-28, Hanoi, Vietnam, 1999 (with R. Baldoni, G. Cioffi, J.-M. Hélary).

[C110] **Solving consensus using Chandra-Toueg's unreliable failure detectors: a general quorum-based approach**. (23/60). *Proc. 13th Int'l Symposium on Distributed Computing (DISC'99)*, Springer LNCS 1693, pp. 49-63, (P. Jayanti Ed.), Bratislava (Slovaquia), 1999 (with A. Mostéfaoui).

[C111] Unreliable failure detector with limited scope accuracy and an application to consensus. (30/84). *Proc. 19th Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS)*, Springer LNCS 1738, pp. 329-340, Chennai (India), 1999 (with A. Mostéfaoui).

[C112] **Computing global functions in asynchronous distributed systems with process crashes**. (76/187). Best Paper Award. *20th IEEE Int'l Conference on Distributed Computing Systems (ICDCS)*, IEEE Computer Press, pp. 584-591, Taipei (Taiwan), 2000 (with J.-M. Hélary, M. Hurfin, A. Mostéfaoui, F. Tronel).

[C113] Atomic broadcast in asynchronous crash-recovery distributed systems. (76/187). *20th IEEE Int'l Conference on Distributed Computing Systems (ICDCS)*, IEEE Computer Press, pp. 288-295, Taipei (Taiwan), 2000 (with L. Rodrigues -Lisbon-).

[C114] Deadline-constrained causal order. *Proc. Third Int'l IEEE Symposium on Object-Oriented Real-Time Distributed Computing (ISORC)*, IEEE Computer Press, pp. 234-241, Newport Beach (CA), 2000 (with E. Anceaume, R. Baldoni, L. Rodrigues).

[C115] Consensus based on failure detectors with a perpetual weak accuracy property. (107/303). *Proc. IEEE Int'l Parallel and Distributed Processing Symposium (IPDPS)*, IEEE Computer Press, pp. .514-519, Cancun (Mexico), 2000 (with A. Mostéfaoui).

[C116] Consensus based on strong failure detectors: time and message efficient protocol. *Proc. Int'l Workshop on Fault-Tolerant Parallel and Distributed Systems, (in conjunction with the Int'l IEEE Conference on Parallel and Distributed Processing Symposium -IPDPS-)*, Springer, LNCS 1800, pp. 1258-1267, Cancun (Mexico), 2000 (with F. Greve, A. Mostéfaoui, R. Macedo).

[C117] The best of both worlds: a hybrid approach to solve consensus. *Int'l IEEE/IFIP Conference on Dependable Systems and Networks (previously IEEE Symposium on Fault-Tolerant Systems)*, IEEE Computer Press, pp. 513-522, New York, 2000 (with A. Mostéfaoui, F. Tronel).

[C118] From crash fault-tolerance to Byzantine fault-tolerance: towards a modular approach. *Int'l IEEE/IFIP Conference on Dependable Systems and Networks (previously IEEE Symposium on Fault-Tolerant Systems)*, IEEE Computer Press, pp. 273-282, New York, June 2000 (with R. Baldoni, J.-M. Hélary).

[C119] Consensus in Byzantine asynchronous distributed systems. *7th Int'l Colloquium on Structural Information and Communication Complexity (SIROCCO)*, Carleton University Press, pp. 1-15, L'Aquila (Italy) 2000 (with R. Baldoni, J.-M. Hélary, L. Tanguy).

[C120] Tracking causality in distributed systems: a suite of efficient protocols. *7th Int'l Colloquium on Structural Information and Communication Complexity (SIROCCO)*, Carleton University Press, pp. 181-195, L'Aquila (Italy), 2000 (with J.-M. Hélary, G. Melideo).

[C121] **k-Set agreement and limited accuracy failure detectors**. *19th ACM SIGACT-SIGOPS Int'l Symposium on Principles of Distributed Computing (PODC'00)*, ACM Press, pp. 143-152, Portland (OR), 2000 (with A. Mostéfaoui).

[C122] Quorum-based replication in asynchronous crash-recovery distributed systems. (167/326). *Proc. 6th Int'l European Parallel Computing Conference (EUROPAR'00)*, Springer LNCS 1900, pp. 554-558, Munich (Germany), 2000 (with L. Rodrigues).

[C123] Logical instantaneity and causal order: two first class communication modes for parallel computing. Invited Talk. *Proc. 6th Int'l European Parallel Computing Conference (EUROPAR)*, Springer LNCS 1900, pp. 13-20, Munich (Germany), 2000.

[C124] **Low-cost consensus-based atomic broadcast**. (21/44). *7th IEEE Pacific Rim Int'l Symposium on Dependable Computing (PRDC'2000)*, IEEE Computer Press, pp. 45-52, UCLA, Los Angeles (CA), 2000. (with A. Mostéfaoui).

[C125] Realtime-based strong consistency for distributed objects. *Proc. 5th Int'l IEEE Workshop on Object-Oriented Real-time Distributed Systems (WORDS'01)*, IEEE Computer Press, pp. 161-168, Roma (Italy), 2001 (with M. Ahamad, F. Torres -Georgia Tech-).

[C126] Primary component asynchronous group membership as an instance of a generic agreement framework. (48/102). *Proc. 5th IEEE Int'l Symposium on Autonomous Decentralized Systems (ISADS)*, IEEE Computer Press, pp. 93-100, Dallas (TX), 2001 (with F. Greve, M. Hurfin , F. Tronel).

[C127] Shared state consistency for time-sensitive distributed applications. (72/210). Best paper Award. *21th IEEE Int'l Conference on Distributed Computing Systems (ICDCS)*, IEEE Computer Press, pp. 606-614, Phoenix (AZ), 2001 (with V. Krishnaswamy, M. Ahamad, D. Bakken -Georgia Tech-).

[C128] A condition for k-set agreement in asynchronous distributed systems. (48/276). *Proc. IEEE Int'l Parallel and Distributed Processing Symposium (IPDPS'01), (15th IPPS/12th SPDP)*, IEEE Computer Press, San Francisco (CA), 2001 (with A. Mostéfaoui).

[C129] Randomized ultivalued consensus. *Proc. Fourth Int'l IEEE Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'01)*, IEEE Computer Press, pp. 195-200, Magdeburg (Germany), 2001 (with P. Ezhilchelvan, A. Mostéfaoui).

[C130] **Conditions on input patterns for consensus solvability in asynchronous distributed systems**. (83/230). *33rd ACM Symposium on Theory of Computing (STOC'01)*, ACM Press, pp. 153-162, 2 Crête (Greece), 2001 (En collaboration avec A. Mostéfaoui, S. Rajsbaum).

[C131] **Randomized k-set agreement**. (34/93). *13th ACM Symposium on Parallel Algorithms and Architectures (SPAA'01)*, ACM Press, pp. 291-297, Crête(Greece), 2001 (with A. Mostéfaoui).

[C132] **A hierarchy of conditions for consensus solvability.** (29/102). *20th ACM SIGACT-SIGOPS Int'l Symposium on Principles of Distributed Computing (PODC'01)*, ACM Press, pp. 151-160, Newport (RI), 2001 (with A. Mostéfaoui, S. Rajsbaum, M. Roy).

[C133] Quiescent uniform reliable broadcast as an introduction to failure detector oracles. (36/81). *6th Int'l Conference on Parallel Computing Technologies (PaCT'01)*, Novosibirsk, Springer LNCS 2127, pp. 98-111, 2001.

[C134] **Consensus in one communication step**. (36/81). *6th Int'l Conference on Parallel Computing Technologies (PaCT'01)*, Novosibirsk, Springer LNCS 2127, pp. 42-50, 2001 (with F. Brasileiro, F. Greve, A. Mostéfaoui).

[C135] Efficient condition-based consensus. *8th Int'l Colloquium on Structural Information and Communication Complexity (SIROCCO'01)*, Carleton University Press, pp. 275-291, Val de Nuria (Catalonia, Spain), 2001(with A. Mostéfaoui, S. Rajsbaum, M. Roy).

[C136] Building TMR-based reliable servers despite bounded input lifetime. *Proc. 7th Int'l European Parallel Computing Conference (EUROPAR'01)*, Springer LNCS 2150, pp. 482-485, Manchester (UK), 2001 (with P. Ezhilchelvan, J.M. Hélary).

[C137] A consensus protocol based on a weak failure detector and a sliding round window. (21/65). *20th IEEE Symposium on Reliable Distributed Systems, (SRDS'01)*, IEEE Computer Press, pp. 120-129, New-Orleans, 200 (with M. Hurfin, R. Macedo, A. Mostéfaoui).

[C138] An adaptive failure detection protocol. (33/80). *8th IEEE Pacific Rim Int'l Symposium on Dependable Computing (PRDC'2001)*, IEEE Computer Press, pp. 146-153, Seoul (South Korea), 2001 (with C. Fetzer, F. Tronel).

[C139] Approximate real-time clocks for scheduled events. *Proc. 5th Int'l IEEE Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'02)*, IEEE Computer Press, pp. 53-61, Washington DC, 2002 (with C. Fetzer) –Dresden University, Germany).

[C140] Wait-free objects for real-time systems (position paper). *Proc. 5th Int'l IEEE Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'02)*, IEEE Computer Press, pp. 413-420, Washington DC, 2002.

[C141] **A versatile and modular consensus protocol**. (48/156). *Int'l IEEE/IFIP Conference on Dependable Systems and Networks (DSN'02), (previously, FTCS)*, IEEE Computer Press, pp. 364-373, Washington DC, 2002 (with A. Mostéfaoui, S. Rajsbaum).

[C142] Ordering vs timeliness: two facets of consistency? Invited Talk. In *Future Directions in Distributed Computing (FuDiCo)*, Springer LNCS 2584, pp. 73-77, Bertinoro (Italy), 2002 (with M. Ahamad).

[C143] Using error-correcting codes to solve distributed agreement problems: a future direction in distributed computing? Invited talk. In *Future Directions in Distributed Computing (FuDiCo)*, Springer LNCS 2584, pp. 17-21, Bertinoro (Italy), 2002 (with R. Friedman, A. Mostéfaoui, S. Rasjbaum).

[C144] Tracking immediate predecessors in distributed computations. *14th ACM Symposium on Parallel Algorithms and Architectures (SPAA'02)*, ACM Press, pp. 210-219, Winnipeg (Canada), 2002 (with E. Anceaume, J.M. Hélary).

[C145] **Asynchronous distributed agreement and its relation with error correcting codes**.(23/75). *Proc. 16th Int'l Symposium on DIStributed Computing (DISC'02)*, Springer LNCS 2508, pp. 63-87, Toulouse (France), 2002 (with R. Friedman, A. Mostéfaoui, S. Rajsbaum).

[C146] The lords of the rings: efficient maintenance of views at data warehouses. (23/75). *Proc. 16th Int'l Symposium on DIStributed Computing (DISC'02)*, Springer LNCS 2508, pp. 33-47, Toulouse ((France), 2002 (with A. Mostéfaoui, M. Roy, A. El Abbadi, D. Agrawal).

[C147] Condition-based protocols for set agreement problems. (23/75) *Proc. 16th Int'l Symposium on DIStributed Computing (DISC'02)*, Springer LNCS 2508, pp. 48-62, Toulouse (France), 2002 (with A. Mostéfaoui, S. Rajsbaum, M. Roy).

[C148] Sequential consistency as lazy linearizability. (119/300). *Proc. First Eurasia ICT-2002: Information and Communication Technology*, Springer LNCS 2510, pp. 866-873, Shiraz (Iran), 2002.

[C149] An introduction to the renaming problem. (29/63). *9th IEEE Pacific Rim Int'l Symposium on Dependable Computing (PRDC'2002)*, IEEE Computer Press, pp. 121-124, Tsukuba (Japan), 2002.

[C150] **Consensus in synchronous systems: a concise guided tour**. (29/63). *9th IEEE Pacific Rim Int'l Symposium on Dependable Computing (PRDC'2002)*, IEEE Computer Press, pp. 221-228, Tsukuba (Japan), 2002.

[C151] An optimal atomic broadcast protocol and implementation framework. *8th Int'l IEEE Workshop on Object-Oriented Real-Time Dependable Systems (WORDS'2003)*, IEEE Computer Press, pp. 32-39, Guadalajara (Mexico), 2003 (with P. Ezhilchelvan –Newcastle University, UK)).

[C152] Reliable compare&swap for fault-tolerant synchronization. Invited Talk. *8th Int'l IEEE Workshop on Object-Oriented Real-Time Dependable Systems (WORDS'2003)*, Guadalaraja (Mexico) IEEE Computer Press, pp. 50-55, 2003.

[C153] Token-based sequential consistency in asynchronous distributed systems. (68/219). *17th Int'l IEEE Conference on Advanced Information Networking and Applications (AINA'03)*, IEEE Computer Press, pp. 421-427, Xi'an (China), 2003.

[C154] Uniform agreement despite process omission failures. Invited Talk. *Proc. Int'l Workshop on Fault-Tolerant Parallel and Distributed Systems, (in conjunction with the Int'l IEEE Conference on Parallel and Distributed Processing Symposium (IPDPS'03)*, IEEE Computer Press, Nice (France), 2003 (En collaboration avec PH. RAIPIN PARVEDY.)

[C155] **A generic framework for indulgent consensus**. (72/406) *23th IEEE Int'l Conference on Distributed Computing Systems (ICDCS'03)*, IEEE Computer Society Press, pp. 88-97, Providence (RI), 2003 (with R. Guerraoui –EPFL, Lausanne–).

[C156] Elastic vector time. (72/406). *23th IEEE Int'l Conference on Distributed Computing Systems (ICDCS'03)*, IEEE Computer Press, pp. 284-291, 2003 (with C. Fetzer –Dresden University, Germany–).

[C157] Looking for a common view for mobile worlds. *9th IEEE Conference on Future Trends of Distributed Computing Systems (FT DCS'03).* IEEE Computer Press, pp. 159-165, Porto Rico, 2003 (with M. Gradinariu, G. Simon).

[C158] Nested Invocation protocol on object-based systems. *Proc. 6th Int'l IEEE Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'03)*, IEEE Computer Society Press, Hakodate (Hokkaido, apan), 2003 (with K. Hori, T. Enokido and M. Takizawa -Tokyo-).

[C159] Anonymous Publish/subscribe in peer-to-peer network systems. *Proc. IEEE Int'l Parallel and Distributed Processing Symposium (IPDPS'03)*,IEEE Computer Press, Nice (France), 2003 (with A. Datta, M. Gradinariu, G. Simon).

[C160] **Asynchronous implementation of failure detectors**. (45/146). *Int'l IEEE Conference on Dependable Systems and Networks (DSN'03)*, (Track: *Dependable Computing and Communications Symposium*), IEEE Computer Press, pp. 351-360, San Francisco (CA), 2003 (with A. Mostéfaoui, E. Mourgaya).

[C161] Evaluating the condition-based approach to solve consensus. (28/79). *Int'l IEEE Conference on Dependable Systems and Networks (DSN'03)*, (Track: *Performance and Dependability Symposium (IPDS)*), IEEE Computer Press, pp. 541-550, San Francisco (CA), 2003 (with A. Mostéfaoui, E. Mourgaya, Ph. Raipin Parvedy).

[C162] The condition-based approach in distributed computing. Invited talk. *10th Int'l Workshop on Expressiveness in Concurrency (Express'03)*, (Satellite Workshop of *CONCUR'03*), Electronic Notes in Theoretical Computer Science, Vol. 3, p. 79, Marseille (France), 2003.

[C163] A hierarchy of conditions for asynchronous interactive consistency. *7th Int'l Conference on Parallel Computing Technologies (PaCT'03)*, Springer LNCS 2763, pp. 130-140, Nizhni Novgorod (Russia), 2003 (with A. Mostéfaoui, S. Rajsbaum, M. Roy).

[C164] The renaming problem as an introduction to structures for wait-free computing. *7th Int'l Conference on Parallel Computing Technologies (PaCT'03)*, Springer LNCS 2763, pp. 151-164, Nizhni Novgorod (Russia), 2003.

[C165] Using conditions to expedite consensus in synchronous distributed systems. (25/91). *Proc. 17th Int'l Symposium on DIStributed Computing (DISC'03)*, Springer LNCS 2848, pp. 249-263, Sorrento (Italy), 2003 (with A. Mostéfaoui, S. Rajsbaum).

[C166] A Distributed implementation of sequential consistency with multi-object operations. (84/475). *24th IEEE Int'l Conference on Distributed Computing Systems (ICDCS'04)*, IEEE Computer Society Press, pp. 544-551, Tokyo, 2004 (with K. Vidyasankar -Newfoundland University, Canada-).

[C167] A hybrid approach for building eventually accurate failure detectors. (32/104). *10th IEEE Pacific Rim Int'l Symposium on Dependable Computing (PRDC'2004)*, IEEE Computer Society Press, pp. 57-65, Papeete (Tahiti, France), 2004 (with A. Mostéfaoui, D. Powell).

[C168] **Optimal early-stopping uniform consensus in synchronous systems with process omission failures**. *16th ACM Symposium on Parallel Algorithms and Architectures (SPAA'04)*, ACM Press, pp. 302-310, Barcelona (Spain), 2004 (with Ph. Raipin Parvedy)

[C169] **The synchronous condition-based consensus hierarchy**. (31/142). *Proc. 18th Int'l Symposium on DIStributed Computing (DISC'04)*, Springer LNCS 3274, pp. 1-15, Amsterdam, 2004 (with A. Mostéfaoui, S. Rajsbaum).

[C170] On the respective power of $\Diamond\mathcal{P}$ and $\Diamond\mathcal{S}$ to solve one-shot agreement problems. (31/142). *Proc. 18th Int'l Symposium on DIStributed Computing (DISC'04)*, Springer LNCS 3274, pp. 41-55, Amsterdam, 2004 (with R. Friedman, A. Mostéfaoui).

[C171] A methodological construction of an efficient sequential consistency protocol. (26/85). *Proc. 3rd IEEE Int'l Symposium on Network Computing and Applications (NCA'04)*, IEEE Computer Press, pp. 141-148, Kendall Square, Cambridge (MA), 2004 (with V. Cholvi, A. Fernandez, E. Jimenez -Madrid-).

[C172] Simple and efficient oracle-based consensus protocols for asynchronous Byzantine systems. (27/117). *23th IEEE Symposium on Reliable Distributed Systems (SRDS'04)*, IEEE Computer Press, pp. 228-237, Florianopolis (Brazil), 2004 (with R. Friedman, A. Mostéfaoui).

[C173] Crash-Resilient Time-free eventual leadership. (27/117). *23th IEEE Symposium on Reliable Distributed Systems (SRDS'04)*, IEEE Computer Society Press, pp. 208-217, Florianopolis (Brazil), 2004 (with A. Mostéfaoui, C. Travers).

[C174] Oracles pour la tolérance aux fautes dans les systèmes répartis. Invited talk. *7ème Colloque Africain sur la Recherche Informatique (CARI'04)*, Hammamet (Tunisie), 2004.

[C175] On the benefits of the functional modular approach in distributed data management systems. Invited talk. *Proc. First IEEE Workshop on Dependable Distributed Data Management (WDDDM'04, SRDS'04 satellite workshop)*, IEEE Computer Society Press, pp. 1-6, Florianopolis (Brazil), 2004 (with R. Friedman –the Technion–).

[C176] The notion of veto number for distributed agreement problems. Invited talk. *6th International Workshop on Distributed Computing (IWDC'04, Now ICDCN)*, Springer LNCS 3326, pp. 315-325, Kolkata (India), 2004 (with R. Friedman, A. Mostéfaoui).

[C177] A simple protocol offering both atomic consistent read operations and sequentially consistent read operations. *18th Int'l IEEE Conference on Advanced Information Networking and Applications (AINA'05)*, IEEE Computer Press, pp. 961-966, Taipei (Taiwan), March 2005 (with M. Roy, C. Tutu).

[C178] **Building and using quorums despite any number of process crashes**. *5th European Dependable Computing Conference (EDCC'05)*, Budapest (Hungary), Springer LNCS 3463, pp. 2-19, 2005 (with R. Friedman, A. Mostéfaoui).

[C179] Mixed consistency model: meeting data sharing needs of heterogeneous users. (75/543). *25th IEEE Int'l Conference on Distributed Computing System (ICDCS'05)*, IEEE Computer Press, pp. 209-218, Columbus (Ohio), 2005 (with M. Ahamad, Z. Zhan -Georgia Tech-).

[C180] Building responsive TMR-based servers in the presence of timing constraints. *Proc. 8th Int'l IEEE Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05)*, IEEE Computer Press, pp. 267-274, Seattle (WA), 2005 (with P. Ezhilchelvan, J.-M. Hélary).

[C181] A hybrid and adaptive model for fault-tolerant distributed computing. *Int'l IEEE Conference on Dependable Systems and Networks (DSN'05)*, IEEE Computer Society Press, pp. 412-421, Yokohama (Japan), 2005 (with S. Gorender, R. Macedo -Salvador de Bahia-).

[C182] Early-stopping k-set agreement in synchronous systems prone to any number of process crashes. *8th Int'l Conference on Parallel Computing Technologies (PaCT'05)*, Krasnoyarsk (Russia), Springer LNCS 3606, pp. 49-58, 2005 (with P. Raipin Parvedy, C. Travers).

[C183] Allowing atomic objects to coexist with sequentially consistent objects. *8th Int'l Conference on Parallel Computing Technologies (PaCT'05)*, Krasnoyarsk (Russia), Springer LNCS 3606, pp. 59-73, Sept. 2005. (With M. Roy.)

[C184] **The combined power of conditions and failure detectors to solve asynchronous set agreement**. (36/160). *24th ACM SIGACT-SIGOPS Int'l Symposium on Principles of Distributed Computing (PODC'05)*, ACM Press, pp. 179-188, Las Vegas (NV), 2005 (with A. Mostéfaoui, S. Rajsbaum).

[C185] From static distributed systems to dynamic systems. (20/67). *24th IEEE Symposium on Reliable Distributed Systems (SRDS'05)*, IEEE Computer Society Press, pp. 109-119, Orlando (FL), 2005 (with A. Mostéfaoui, C. Travers, S. Peterson, A. El Abbadi, D. Agrawal (-UCSB-).

[C186] Intersecting sets: a basic abstraction for asynchronous agreement problems. (38/105). *11th IEEE Pacific Rim Int'l Symposium on Dependable Computing (PRDC'2005)*, IEEE Computer Press, pp. 15-22, Changsha (Hunan), (China), 2005 (with R. Friedman, A. Mostéfaoui).

[C187] Decision optimal early-stopping k-set agreement in synchronous systems prone to send omission failures. (38/105). *11th IEEE Pacific Rim Int'l Symposium on Dependable Computing (PRDC'2005)*, IEEE Computer Society Press, pp. 23-30, Changsha (Hunan), (China), 2005 (with P. Raipin Parvedy, C. Travers).

[C188] A note on a simple equivalence between round-based synchronous and asynchronous models. (38/105). *11th IEEE Pacific Rim Int'l Symposium on Dependable Computing (PRDC'2005)*, IEEE Computer Society Press, pp. 387-390, Changsha (Hunan), (China), 2005 (with M. Roy).

[C189] Fault-tolerant techniques for concurrent objects. *2nd Latin-American Symposium on Dependable Computing*, Springer LNCS 3747, pp. 265, 2005 (with R. Guerraoui).

[C190] Abstractions for implementing atomic objects in distributed systems. (30/109). *9th Int'l Conference on Principles of Distributed Systems (OPODIS'05)*, Springer LNCS 3974, pp. 73-87, Pisa (Italy), D2005 (with R. Friedman, C. Travers).

[C191] A hierarchical consensus protocol for mobile adhoc networks. *14th Euromicro Int'l Conference on Parallel, Distributed and Network-based Processing (PDP'06)*, IEEE Computer Society Press, pp. 64-71, Montbéliard-Sochaux (France), 2006 (with W. Wu, J. Cao, J. Yang -Hong-Kong-).

[C192] **The committee decision problem**. (66/224). *Proc. Latin American Theoretical Informatics Symposium (LATIN'06).* Springer LNCS 3887, pp. 502-514, 2006 (with E. Gafni, S. Rajsbaum, C. Travers).

[C193] From failure detectors with limited scope accuracy to system wide leadership. (153/521). *19th Int'l IEEE Conference on Advanced Information Networking and Applications (AINA'06)*, IEEE Computer Society Press, pp. 81-86, Vienna (Austria),2006 (with A. Mostéfaoui, S. Rajsbaum, C. Travers).

[C194] A leader election protocol for eventually synchronous shared memory systems. *4th Int'l IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS'06)*, IEEE Computer Press, pp. 75-80, Gyeongju (South Korea), 2006 (with R. Guerraoui).

[C195] **Leader election with weak assumptions on initial knowledge, communication reliability and synchrony**. (34/187) *36th Int'l IEEE Conference on Dependable Systems and Networks (DSN'06)*, IEEE Computer Society Press, pp. 166-175, Philadelphia (Pennsylvania), June 2006 (with A. Fernandez, E. Jimenez -Madrid-).

[C196] Strongly-terminating early-stopping k-set agreement in synchronous systems with general omission failures. *13th Colloquium on Structural Information and Communication Complexity (SIROCCO'06)*, Springer LNCS 4056, pp. 182-196, Chester (UK), 2006 (with P. Raipin Parvedy, C. Travers).

[C197] From static distributed systems to dynamic systems: an approach for a first step. Invited Talk. *Proc. International workshop on Dynamic Distributed Systems (IWDDS'06)*, in conjunction with *25th IEEE Int'l Conference on Distributed Computing Systems (ICDCS'06)*, IEEE Computer Society Press, Lisbon (Portugal), 2006.

[C198] **Irreducibility and additivity of set agreement-oriented failure detectors**. *25th ACM SIGACT-SIGOPS Int'l Symposium on Principles of Distributed Computing (PODC'06)*, ACM Press, pp. 153-162, Denver (Colorado), 2006 (with A. Mostéfaoui, S. Rajsbaum, C. Travers).

[C199] The power and limit of adding synchronization messages for synchronous agreement. (64/201) *Proc. 35th Int'l Conference on Parallel Processing (ICPP'06)*, IEEE Computer Society Press, pp. 399-406, Colombus (Ohio), 2006 (with J. Cao, W. Wu, X. Wang -Hong-Kong-).

[C200] **Exploring Gafni's reduction land: from Ω^k to wait-free adaptive $(2p-\lceil \frac{p}{k} \rceil)$-renaming via k-set agreement**. (35/145) *Proc. 20th Int'l Symposium on DIStributed Computing (DISC'06)*, Springer LNCS 4167, pp. 1-16, Stockholm (Sweden), 2006 (with A. Mostéfaoui, C. Travers).

[C201] Synchronous set agreement: a concise guided tour (including a new algorithm and a list of open problems). (41/117 *Proc. 12th Int'l IEEE Pacific Rim Dependable Computing Symposium (PRDC'2006)*, IEEE Society Press, pp. 267-274, Riverside (CA), 2006 (with C. Travers).

[C202] Core persistence in peer-to-peer systems: relating size to lifetime. *Proc. Workshop on Reliability in Decentralized Distributed Systems (RDDS'06)*, Springer LNCS 4278, pp. 1470-1479, Montpellier (France), 2006(with V. Gramoli, Anne-Marie Kermarrec, A. Mostéfaoui, B. Sericola).

[C203] On the fly estimation of the processes that are alive/crashed in an asynchronous message-passing system. (41/117). *Proc. 12th Int'l IEEE Pacific Rim Dependable Computing Symposium (PRDC'2006)*, IEEE Society Computer Press, pp. 257-266, Riverside (CA), 2006 (with A. Mostéfaoui, G. Tredan).

[C204] **Simultaneous consensus tasks: a tighter characterization of set consensus**. (31 regular/245). *Proc. 8th Int'l Conference on Distributed Computing and Networking (ICDCN'06)*, Springer LNCS 4308, pp. 331-341, Guwahati (India), 2006 (with Y. Afek, E. Gafni, S. Rajsbaum, C. Travers).

[C205] In search of the holy grail: looking for the weakest failure detector for wait-free set agreement. Invited Talk. *Proc. 10th Int'l Conference On Principles Of Distributed Systems (OPODIS'06)*, Springer LNCS 4305, pp. 1-17, 2006 (with C. Travers).

[C206] A universal construction for wait-free objects. *Proc. ARES 2007 Workshop on Foundations of Fault-tolerant Distributed Computing (FOFDC 2007)*, IEEE Society Computer Press, pp. 959-966, 2007 (with R. Guerraoui –EPFL, Lausanne–).

[C207] A timing assumption and a t-resilient protocol for implementing an eventual leader service in asynchronous shared memory systems. *Proc. 10th Int'l IEEE Symposium on Objects and Component-oriented Real-time Computing (ISORC 2007)*, IEEE Society Computer Press, pp. 71-78, Santorini (Greece), 2007(with A. Fernandez, E. Jimenez, G. Tredan).

[C208] **The notion of a timed register and its application to indulgent synchronization**. (37/130) *19th ACM Symposium on Parallel Algorithms and Architectures (SPAA'07)*, San Diego (CA), ACM Press, pp. 200-209, 2007 (with G. Taubenfeld).

[C209] From renaming to k-set agreement. *14th Colloquium on Structural Information and Communication Complexity (SIROCCO'07)*, Springer LNCS 4474, pp. 62-76, Castiglioncello (Italy), 2007 (with A. Mostéfaoui, C. Travers).

[C210] **Electing an eventual leader in an asynchronous shared memory system**. *37th Int'l IEEE Conference on Dependable Systems and Networks (DSN'07)*, IEEE Computer Society Press, pp. 399-408, Edinburgh (UK), 2007 (with A. Fernandez, E. Jimenez -Madrid-).

[C211] Distributed slicing in dynamic systems. (71/528) *27th IEEE Int'l Conference on Distributed Computing Systems (ICDCS'07)*, IEEE Computer Society Press, pp. 209-218, Toronto (Canada), 2007(with A. Fernandez, V. Gramoli, E. Jimenez and Anne-Marie Kermarrec).

[C212] Eventual leader service in unreliable asynchronous systems: why? How? Invited Talk. *Proc. 6th IEEE International Symposium on Network Computing and Applications (NCA'07)*, IEEE Computer Society Press, pp. 11-21, Cambridge (MA), 2007.

[C213] From unreliable objects to reliable objects: the case of atomic registers and consensus. *9th Int'l Conference on Parallel Computing Technologies (PaCT'07)*, Peereslavl-Zalessky (Russia), Springer LNCS LNCS 4671, pp. 47-61, 2007 (with R. Guerraoui).

[C214] Towards a definition of dynamic distributed systems. *9th Int'l Conference on Parallel Computing Technologies (PaCT'07)*, Peereslavl-Zalessky (Russia), Springer LNCS 4671, pp. 1-14, 2007 (with R. Baldoni, S. Tucci –La Spienza, University of Roma–,

M. Bertier).

[C215] A subjective visit to selected topics in distributed computing. Invited Talk. *Proc. 21th Int'l Symposium on Distributed Computing (DISC'07)*, Springer LNCS 4731, pp. 5-6, 2007.

[C216] **Test&set, adaptive renaming and set agreement: a guided visit to asynchronous computability**.(25/185). *26th IEEE Symposium on Reliable Distributed Systems (SRDS'07)*, IEEE Computer Society Press, pp. 93-102, Beijing (China), 2007 (with E. Gafni, C. Travers).

[C217] The eventual clustered oracle and its application to consensus in MANETs. (25/185). *26th IEEE Symposium on Reliable Distributed Systems (SRDS'07)*, IEEE Computer Society Press, pp. 23-32, Beijing (China), 2007 (with Jiannong Cao, Weigang Wu -Hong-kong Polytechnic University-).

[C218] The eventual leadership in dynamic mobile networking environments. *Proc. 13th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'07)*, IEEE Computer Society Press, pp. 123-130, Melbourne (Australia), 17-19, 2007 (with Jiannong Cao, Weigang Wu -Hong-kong Polytechnic University-, C. Travers).

[C219] A dual-token-based fault-tolerant mutual exclusion algorithm for MANETs. *Proc. 3rd Int'l Conference on Mobile Ad-hoc and Sensor Networks (MSN 2007)*, Springer LNCS LNCS 4864, pp. 572-583, Beijing (China), 2007 (with Jiannong Cao, Weigang Wu -Hong-kong Polytechnic University-).

[C220] Small-world networks: is there a mismatch between theory and practice? *Proc. 11th Int'l Conference On Principles Of Distributed Systems (OPODIS'07)*, Springer LNCS 4878, pp. 372-385, 2007 (with F. Bonnet, A.-M. Kermarrec).

[C221] **From an intermittent rotating star to a leader**. *Proc. 11th Int'l Conference On Principles Of Distributed Systems (OPODIS'07)*, Springer LNCS 4878, pp. 189-203, 2007 (with A. Fernandez).

[C222] Timed quorum systems for large scale and dynamic environments. *Proc. 11th Int'l Conference On Principles Of Distributed Systems (OPODIS'07)*, Springer LNCS 4878, pp. 429-442, 2007 (with V. Gramoli).

[C223] **Narrowing power vs efficiency in synchronous set agreement**. (30/185). *Proc. 9th Int'l Conference on Distributed Computing and Networking (ICDCN'08)*, Springer LNCS 4904, pp. 99-111, Kolkata (India), 2008 (with A. Mostéfaoui, C. Travers).

[C224] Synchronization is coming back, but is it the same? Invited Talk. *IEEE 22nd Int'l Conference on Advanced Information Networking and Applications (AINA'08)*, pp. 1-10, Okinawa (Japan), 2008.

[C225] **Conditions for set agreement with an application to synchronous systems**. (102/638). *28th IEEE Int'l Conference on Distributed Computing Systems (ICDCS'08)*, IEEE Computer Society Press, pp. 663-672, Beijing (China), 2008 (with F. Bonnet).

[C226] **The iterated restricted immediate snapshot (IRIS) model**. (66/172). *14th Int'l Computing and Combinatorics Conference (COCOON'08)*, Dalian (China), Springer LNCS 5092, pp.487-496, 2008. (with S. Rajsbaum, C. Travers).

[C227] On modeling fault-tolerance of gossip-based reliable multicast protocols. *37th Int'l Conference on Parallel Processing (ICPP'08)*, IEEE Computer Society Press, pp. 149-156, Portland (OR), 2008 (with Xiaopeng Fan, Jiannong Cao, Weigang Wu -Hong-kong Polytechnic University-).

[C228] **No double discount: condition-based simultaneity yields limited gain**. *Proc. 22th Int'l Symposium on Distributed Computing (DISC'08)*, Springer LNCS 5218, pp. 423-437, 2008 (with Y. Moses -The Technion-).

[C229] Locks considered harmful: a look at non-traditional synchronization. *Proc. 6th Int'l Workshop on Embedded and Ubiquitous computing Systems (SEUS'08)*, Springer, LNCS 5287, pp. 369-380, 2008.

[C230] On the solvability of anonymous partial grid exploration by mobile robots. (30 regular/102). *Proc. 12th Int'l Conference On Principles Of Distributed Systems (OPODIS'08)*, Springer LNCS 5401, pp. 428-445, 2008. (with R. Baldoni, F. Bonnet, A. Milani).

[C231] A lock-based STM protocol that satisfies opacity and progressiveness. (30 regular/102). *Proc. 12th Int'l Conference On Principles Of Distributed Systems (OPODIS'08)*, Springer LNCS 5401, pp. 226-245, 2008 (with D. Imbs).

[C232] **Provable STM properties: leveraging clock and locks to favor commit and early abort**. (24 regular/176). *Proc. 10th Int'l Conference on Distributed Computing and Networking (ICDCN'09)*, Springer LNCS 5408, pp. 67-78, Hyderabad (India), 2009 (with D. Imbs).

[C233] Large scale networked systems: from anarchy to geometric self-structuring. (24 regular/176). *Proc. 10th Int'l Conference on Distributed Computing and Networking (ICDCN'09)*, Springer LNCS 5408, pp. 25-36, Hyderabad (India), 2009. (with A.-M. Kermarrec, A. Mostéfaoui, A. Viana, G. Trédan).

[C234] Shared memory synchronization in the presence of failures: an exercise-based introduction for the sophomore. *IEEE Int'l Conference on Complex, Intelligent and Software Intensive Systems (CISIS'09)*, IEEE Computer Society Press, pp. 9-18, Fukuoka (Japan), 2009.

[C235] **Implementing a register in a dynamic distributed system.** (74/455). *29th IEEE Int'l Conference on Distributed Computing Systems (ICDCS'09)*, IEEE Computer Society Press, pp. 639-647, Montreal (Canada), June 2009. (With R. Baldoni, S., Bonomi, A.-M. Kermarrec).

[C236] **A versatile STM protocol with invisible read operations that satisfies the virtual world consistency condition.** *Proc. 16th Colloquium on Structural Information and Communication Complexity (SIROCCO'09)*, Springer LNCS, 5869, pp. 266-280, May 2009. (With D. Imbs).

[C237] Regular register: an implementation in a churn prone environment. *16th Colloquium on Structural Information and Communication Complexity (SIROCCO'09)*, Springer LNCS, 5869, pp. 15-29, May 2009. (With R. Baldoni, S., Bonomi).

[C238] Software transactional memories: an approach for multicore programming. *10th Int'l Conference on Parallel Computing Technologies (PaCT'09)*, Novosibirsk (Russia), Springer LNCS 5698, pp. 26-40, 2009. (With D. Imbs).

[C239] Adding dynamicity to the uncertainty that characterizes distributed systems: challenges ahead. *Proc. Franco-Brazilian Colloquium on Advances and Challenges in Computer Science (COLIBRI'09)*, pp. 141-147, Bento Gonçalves (Brazil), july 2009. (With R. Macedo, Salvador de Bahia, Brazil).

[C240] Software transactional memory: what? why? how? a new challenge? Invited Talk. *Proc. Franco-Brazilian Colloquium on Advances and Challenges in Computer Science (COLIBRI'09)*, pp. 5-7, Bento Gonçalves (Brazil), july 2009.

[C241] How to implement a shared memory in a dynamic system? Which are the constraints? Invited Talk. *Int'l Workshop on Theoretical Aspects of Dynamic Distributed Systems (TADDS'09)*, in conjunction with DISC 2009, Elche (Spain), 2009.

[C242] **Help when needed, but no more: efficient read/write partial snapshot.** (33/117). *Proc. 23th Int'l Symposium on Distributed Computing (DISC'09)*, Springer LNCS 5805, pp. 142-156, 2009 (With D. Imbs).

[C243] **The price of anonymity: optimal consensus despite asynchrony, crash and anonymity**.(33/117). *Proc. 23th Int'l Symposium on Distributed Computing (DISC'09)*, Springer LNCS 5805, pp. 341-355, 2009 (With F. Bonnet).

[C244] **Looking for the weakest failure detector for k-Set agreement in message-passing systems: Is Π_k the end of the road?** Best paper Award. (49/126). *Proc. 11th Int'l Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'09)*, Springer LNCS 5873, pp. 149-164, 2009 (With F. Bonnet).

[C245] Visiting Gafni's reduction land: from the BG simulation to the extended BG simuation. (49/126). *Proc. 11th Int'l Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'09)*, Springer LNCS 5873, pp. 369-383, 2009 (With D. Imbs).

[C246] Joining a distributed shared memory computation in a dynamic distributed system. *Proc. 7th Workshop on Software Technologies for Future Embedded and Ubiquitous Computing Systems (SEUS'09)*, Springer LNCS 5860, pp. 91-102, 2009 (With R. Baldoni, S. Bonomi).

[C247] D2HT: the best of both worlds, integrating RPS (random peer sampling) and DHT. *8th European Dependable Computing Conference (EDCC'10).* IEEE Computer Society Press, Valencia (Spain), April 2010 (With M. Bertier, F. Bonnet, A.-M. Kermarrec, V. Leroy, S. Peri).

[C248] Early consensus in message-passing systems enriched with a perfect failure detector and its application in the Theta model. *8th European Dependable Computing Conference (EDCC'10).* IEEE Computer Society Press, Valencia (Spain), April 2010 (With F. Bonnet).

[C249] Consensus in anonymous distributed systems: Is there a weakest failure detector? *24th IEEE International Conference on Advanced Information Networking and Applications (AINA'10)*, IEEE Computer Society Press, Perth (Australia), April 2010 (With F. Bonnet).

[C250] **The multiplicative power of consensus numbers**. (39/179). *29th ACM Symposium on Principles of Distributed Computing (PODC'10)*, ACM Press, pp. 26-35,July 2010, Zurich. (With D. Imbs.)

[C251] **On asymmetric progress conditions** .(39/179). *29th ACM Symposium on Principles of Distributed Computing (PODC'10)*, ACM Press, pp. 55-64, July 2010, Zurich. (With D. Imbs, G. Taubenfeld.)

[C252] Value-based sequential consistency for set objects in dynamic distributed systems. *Proc. 16th Int'l European Parallel Computing Conference (EUROPAR'10)*, Springer LNCS 6271, pp. 523-534, 2010. (With R. Baldoni, S., Bonomi).

[C253] The x-wait-freedom progress condition. *Proc. 16th Int'l European Parallel Computing Conference (EUROPAR'10)*, Distinguished paper Award. Springer LNCS 6271, pp. 584-595, 2010. (With D. Imbs.)

[C254] **Anonymous asynchronous systems: the case of failure detectors**. Best paper Award. *Proc. 24th Int'l Symposium on Distributed Computing (DISC'10)*, Springer LNCS 6343, pp. 206-220, 2010. (With F. Bonnet.)

[C255] On adaptive renaming under eventually limited contention. *Proc. 12th Int'l Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'10)*, Springer LNCS 6366, pp. 377-387, 2010. (With D. Imbs.)

[C256] Signature-free broadcast-based intrusion tolerance: never decide a Byzantine value. *Proc. 14th Int'l Conference On Principles Of Distributed Systems (OPODIS'010)*, Springer LNCS 6490, pp. 144-159, 2010. (With A. Mostéfaoui.)

[C257] **A necessary and sufficient condition for solving Byzantine consensus in symmetric networks**. (31 regular/140). *Proc. 12th Int'l Conference on Distributed Computing and Networking (ICDCN'11)*, Springer LNCS 6522, pp. 215-226, Bangalore (India), 2011. (With O. Baldellon, A. Mostéfaoui.)

[C258] A theory-oriented introduction to wait-free synchronization based on the adaptive renaming problem. *IEEE 25nd Int'l Conference on Advanced Information Networking and Applications (AINA'11)*, IEEE Computer Press, pp. 356-363, Singapore, March 2011. (With S. Rajsbaum.)

[C259] A simple snapshot algorithm for multicore systems. *Proc. 5th IEEE Latin-American Symposium on Dependable Computing (LADC'11)*, IEEE Computer Press, pp. 17-23, Sao José dos Campos (Brazil) March 2011. (With D. Imbs.)

[C260] k-Bounded set objects in eventually synchronous distributed systems with churn and continuous accesses. *Proc. 13th European Workshop on Dependable Computing (EDCC'11)*; ACM Digital Library, ISBN 978-1-4503-0284-5, 2011. (With R. Baldoni, S. Bonomi –Universitá L Sapienza, Roma.)

[C261] Looking for efficient implementations of concurrent objects. *11th Int'l Conference on Parallel Computing Technologies (PaCT'11)*, Kazan, (Russia), Springer LNCS 6873, pp. 74-87, 2011. (With A. Mostéfaoui).

[C262] Power and limits of distributed computing shared memory models. Invited talk. *Proc. 2nd Int'l Workshop on Logical Aspects of Fault-Tolerance (LAFT'11)*, in conjunction with the Int'l conference *Logic In Computer Science (LICS'11)*, Toronto, Canada, June 2011. (With S. Rajsbaum –UNAM, Mexico–.)

[C263] Specifying and implementing an eventual leader service for dynamic systems. (56/168). *Proc. 14th Int'l Conference on Network-Based Information Systems (NBiS'11)*, pp. 243-249, IEEE Press, September 2011. (With M. Larrea –University of the Basque Country, San Sebastian, Spain–.)

[C264] Distributed computing with mobile robots: an introductory survey. (56/168). *Proc. 14th Int'l Conference on Network-Based Information Systems (NBiS'11)*, pp. 318-324, IEEE Press, September 2011. (With M. Gradinariu, S. Tixeuil –LIP6, Paris VI–.)

[C265] **The universe of symmetry breaking tasks**. *Proc. 18th Int'l Colloquium on Structural Information and Communication Complexity (SIROCCO'11)*, Springer LNCS 6796, pp. 66-77, 2011. (With D. Imbs, S. Rajsbaum –UNAM, Mexico–.)

[C266] A survey on some recent advances in shared memory models. (56/168). *Proc. 18th Int'l Colloquium on Structural Information and Communication Complexity (SIROCCO'11)*, Springer LNCS 6796, pp. 17-28, 2011. (With S. Rajsbaum –UNAM, Mexico–.)

[C267] Ressources informatiques : encore une histoire de temps ! *Colloque du 20ème Anniversaire de l'Institut Universitaire de France*, Les colloques de l'IUF, Presses de l'université de St-Etienne, pp. 225-243, Lyon, Mai 2011. (With Benoit A., Paschos V., Robert Y., and Trystram D.)

[C268] On the implementation of concurrent objects. *Dependable and Historic Computing (Randell's Tales: a Festschrift recognising the contributions of Brian Randell)*, Springer LNCS 6875, pp. 453-478, 2011.

[C269] **The weakest failure detector to implement a register in asynchronous systems with hybrid communication**. (29/79). Best paper Award. *Proc. 13th Int'l Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'11)*, Springer LNCS 6976, pp. 268-282, 2011. (With D. Imbs.)

[C270] Relations linking failure detectors associated with k-set agreement in message-passing systems. (29/79). *Proc. 13th Int'l Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'11)*, Springer LNCS 6976, pp. 341-355, 2011. (With A. Mostéfaoui, J. Stainer.)

[C271] Read invisibility, virtual world consistency and permissiveness are compatible. (24/88). *Proc. 11th Int'l Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*, Springer LNCS 7016, pp. 245-258, 2011. (With T. Crain, D. Imbs.)

[C272] Towards a universal construction for transaction-based multiprocess programs. *Proc. 13th Int'l Conference on Distributed Computing and Networking (ICDCN'12)*, Springer LNCS 7129, pp. 61-75, Hong-Kong, 2012. (With T. Crain, D. Imbs.)

[C273] A transaction-friendly binary search tree. (26/175). *Proc. 17th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP'12)*, ACM Press, pp. 161-170, 2012. (With T. Crain, V. Gramoli.)

[C274] A look at distributed recursion. Invited talk. *LADA (Languages for Distributed Algorithms), Satellite Workshop of the 33rd ACM POPL (Principles of Programming Languages) conference*, ACM Press, January 2012.

[C275] **Renaming is weaker than set agreement but for perfect renaming: A Map of Sub-Consensus Tasks**. *Proc. 10th Latin American Theoretical INformatics Symposium (LATIN'12)*, Springer LNCS 7256, pp. 145-156, 2012. (With A. Castaneda, D. Imbs, S. Rajsbaum–UNAM, Mexico–.)

[C276] Trying to unify the LL/SC synchronization primitive and the notion of a timed register. (126/445). *IEEE 26nd Int'l Conference on Advanced Information Networking and Applications (AINA'12)*, IEEE Computer Press, pp. 326-330, Fukuoka, March 2012. (With D. Imbs.)

[C277] **Failure detectors in homonymous distributed systems (with an application to consensus)**. (71/515). *Proc. 32nd Int'l Conference on Distributed Computing Systems (ICDCS'12)*, IEEE Computer Press, pp. 275-284, 2012. (With S. Arevalo, A. Fernandez, D. Imbs, E. Jimenez –Madrid–.)

[C278] A simple asynchronous shared memory consensus algorithm based on Ω and closing sets. (54/215). *Proc. 6th Int'l Conference on Complex, Intelligent, and Software Intensive Systems (CISIS'12)*, IEEE Computer Press, pp. 357-364, 2012. (With J. Stainer.)

[C279] Leader election: from Higham-Przytycka's algorithm to a gracefully degrading algorithm. (54/215). *Proc. 6th Int'l Conference on Complex, Intelligent, and Software Intensive Systems (CISIS'12)*, IEEE Computer Press, pp. 225-232, 2012. (With José-Ramon Mendivil and Itziar Arrieta – Pamplona University, Spain–.)

[C280] **Increasing the power of the iterated immediate snapshot model with failures detectors**. *Proc. 19th Int'l Colloquium on Structural Information and Communication Complexity (SIROCCO'12)*, Springer LNCS 7355, pp. 231-242, 2012. (With J. Stainer.)

[C281] **From a store-collect object and Ω to efficient asynchronous consensus**. *Proc. 18th Int'l European Parallel Computing Conference (EUROPAR'12)*, Springer LNCS 7484, pp. 427-438, 2012.(With J. Stainer.)

[C282] **Chasing the weakest failure detector for k-set agreement in message-passing systems**. *Proc. 11th IEEE Int'l Symposium on Network Computing and Applications (NCA'12)*, IEEE Press, pp. 44-51, Boston (USA), 2012. (With A. Mostéfaoui, J. Stainer.)

[C283] STM systems: enforcing strong isolation between transactions and non-transactional code. *Proc. Int'l Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'12)*, Springer LNCS 7439, pp. 317-331, 2012. (With T. Crain, E. Kanellou.)

[C284] When and how process groups can be used to reduce the renaming Space. *Proc. 16th Int'l Conference On Principles Of Distributed Systems (OPODIS'12)*, Springer LNCS 7702, pp. 91–105 , 2012. (With A. Castaneda, J. Stainer.)

[C285] Coordination and computation in distributed intelligent MEMS.(159/567). *IEEE 27nd Int'l Conference on Advanced Information Networking and Applications (AINA'13)*. IEEE Computer Press, pp.129–136, Barcelona, March 2013. (With J. Bourgeois J., Cao J., Raynal M., Dhoutaut D., Piranda, E. Dedu E., Mostéfaoui M., and Mabed H.)

[C286] A short introduction to synchronous communication. (159/567). *IEEE 27nd Int'l Conference on Advanced Information Networking and Applications (AINA'13)*, IEEE Computer Press, pp. 1136-1143, Barcelona, March 2013.

[C287] **Agreement via symmetry breaking: On the structure of weak subconsensus tasks**. (106/494). *Proc. 27th IEEE Int'l Parallel & Distributed Processing Symposium (IPDPS'13)*, IEEE Press, pp. 1147-1158, 2013. (With A. Castaneda, S. Rajsbaum –UNAM, Mexico–.)

[C288] **No hot spot non-blocking skip list**. (61/464). *Proc. 33nd Int'l Conference on Distributed Computing Systems (ICDCS'13)*, IEEE Computer Press, pp. 196-205, 2013. (With T. Crain, V. Gramoli.)

[C289] On the consensus number of non-adaptive perfect renaming. *Proc. First Int'l Conference on Networked Systems (NETYS'13)*, Springer LNCS 7853, pp. 1-12, 2013. (With A. Castaneda – UNAM, Mexico–.)

[C290] **Round-based synchrony weakened by message adversaries vs asynchrony enriched with failure detectors.** *32th ACM Symposium on Principles of Distributed Computing (PODC'13)*, ACM Press, pp. 166-175, 2013. (With J. Stainer.)

[C291] A Contention-friendly binary search tree. (70/261). *Proc. 19th Int'l European Parallel Computing Conference (EUROPAR'13)*, Springer LNCS 8097, pp. 229-240, 2013. (With T. Crain, V. Gramoli.)

[C292] Cliff-edge consensus: agreeing on the precipice. *12th Int'l Conference on Parallel Computing Technologies (PaCT'13)*, St-Petersbourg (Russia), Springer LNCS 7979, pp. 51-64, 2013. (With Taïnai F., Porter B., Coulson G.)

[C293] Simultaneous consensus vs set agreement a message-passing sensitive hierarchy of agreement problems. *Proc. 20th Int'l Colloquium on Structural Information and Communication Complexity (SIROCCO'13)*, Springer LNCS 8179, pp. 298-309, 2013. (With J. Stainer.)

[C294] A generalized mutual exclusion problem and its algorithm. *42nd Int'l Conference on Parallel Processing (ICPP'13)*, IEEE Press pp. 300-309, Lyon (France), 2013. (With A. Luo and W. Wu –Sun Yat-Sen University, Guangzhou, China–, and J. Cao –Poly U, Hong-Kong–.)

[C295] Fault-tolerant leader election in mobile dynamic distributed systems. *19th IEEE Int'l Pacific Rim Symposium on Dependable Computing*. IEEE Press, pp. 78-87, 2013. (With Gómez-Calzado, Lafuente A., Larrea M., Spain.)

[C296] Concurrency-related distributed recursion. Invited talk. *Proc. 15th Int'l Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'13)*, Springer LNCS 8255, pp. xviii-xx, 2013.

[C297] **Computing in the presence of concurrent solo executions**. *Proc. 11th Latin American Symposium on Theoretical INformatics (LATIN'14)*, Springer LNCS 8392, pp. 214-225, 2014. (With M. Herlihy, S. Rajsbaum, J. Stainer).

[C298] What can be computed in a distributed system? Invited talk. *Workshop "From Programs to Systems: The Systems Perspective in Computing" workshop in honor of Professor Joseph Sifakis*, Springer LNCS 8415, pp. 209-224, 2014.

[C299] A simple broadcast algorithm for recurrent dynamic systems. *IEEE 28nd Int'l Conference on Advanced Information Networking and Applications (AINA'14)*. IEEE Computer Press, pp. 933-939, Vancouver, May 2014. (With Cao J. –Hong Kong Polytechnic University–, J. Stainer –IRISA–, W. Wu –Sun Yat-Sen Univerity, Guangzhou, China–.)

[C300] Simple deadlock detection for the And-communication model. (50/167.) *8th IEEE Int'l Conference on Complex, Intelligent and Software Intensive Systems (CISIS'14)*, IEEE Computer Society Press, pp. 273-278, Birmingham (UK), 2014.

[C301] **Signature-free asynchronous Byzantine consensus with $t < n/3$ and $O(n^2)$ messages.** Best paper Award. *Proc. 33th ACM Symposium on Principles of Distributed Computing (PODC'14)*, ACM Press, pp. 2-9, 2014. (With A. Mostéfaoui and Hamouma Moumen.)

[C302] From Turing to the clouds (on the computability power of distributed systems). Invited talk. *Proc. 21th Int'l Colloquium on Structural Information and Communication Complexity (SIROCCO'14)*, Springer LNCS 8576, pp. xiii-xiv, 2014.

[C303] **Reliable shared memory abstractions on top of asynchronous Byzantine message-passing systems**. *Proc. 21th Int'l Colloquium on Structural Information and Communication Complexity (SIROCCO'14)*, Springer LNCS 8576, pp. 37-53, 2014. (With D. Imbs, S. Rajsbaum, J. Stainer.)

[C304] An Exercise in concurrency: from non-blocking objects to fair objects. (52/181). *17th IEEE Int'l Conference on Network-based Information Systems (NBIS'14)*, IEEE Computer Society Press, pp. 1-7, 2014. (With C. Delporte and H. Fauconnier.)

[C305] Fair synchronization in the presence of process crashes, and its weakest failure detector. *Proc. 33th Int'l Symposium on Reliable Distributed Systems (SRDS'14)*, IEEE Press, pp. 161-170, 2014. (With C. Delporte, H. Fauconnier.)

[C306] **Distributed universality**. (32/98). *Proc. 18th Int'l Conference On Principles Of Distributed Systems (OPODIS'14)*, Springer LNCS 8878, pp. 469484, 2014. (With J. Stainer and G. Taubenfeld.)

[C307] A simple predicate to expedite the termination of a randomized consensus algorithm (140/472). *IEEE 29nd Int'l Conference on Advanced Information Networking and Applications (AINA'15)*. IEEE Computer Press, pp. 106-111, March 2015.

[C308] Fisheye consistency: keeping data in synchr in a georeplicated world. *Proc. Third Int'l Conference on Networked Systems (NETYS'15)*, Springer LNCS 9466, pp. 246-262, 2015. (With R. Friedman and F. Taïani.)

[C309] **Minimal synchrony for Byzantine consensus**. *Proc. 34th ACM Symposium on Principles of Distributed Computing (PODC'15)*, ACM Press, pp. 461-470, 2015. (With Z. Bouzid and A. Mostéfaoui.)

[C310] **Stabilizing server-based storage in Byzantine asynchronous message-passing systems**. *Proc. 34th ACM Symposium on Principles of Distributed Computing (PODC'15)*, ACM Press, pp. 471-480, 2015. (With S. Bonomi (Roma), M. Potop-Butucaru (LIP6), and S. Dolev (Ben Gourion University, Israël).)

[C311] Communication patterns and input patterns in distributed computing. Invited talk. *Proc. 22th Int'l Colloquium on Structural Information and Communication Complexity (SIROCCO'15)*, Springer LNCS 9439, pp. 1-15, 2015.

[C312] Concurrent systems: hybrid object implementations and abortable objects. Invited talk. *Proc. 21th Int'l European Parallel Computing Conference (EUROPAR'15)*, Springer LNCS 9233, pp. 3-15, 2015.

[C313] **Signature-free asynchronous Byzantine systems: from multivalued to binary consensus with $t < n/3$, $O(n^2)$ messages, and constant time.** *Proc. 22nd Int'l Colloquium on Structural Information and Communication Complexity (SIROCCO'15)*, Springer LNCS 9439, pp. 194-208, 2015. (With A. Mostéfaoui.)

[C314] Parallel computing *vs* distributed computing: a great confusion? *Proc. 1st European Workshop on Parallel and Distributed Computing Education for Undergraduate Students (Euro-EDUPAR), Satellite workshop of EUROPAR'15*, Springer LNCS 9523, pp. 41-53, 2015.

[C315] **Specifying concurrent problems: beyond linearizability and up to tasks**. *Proc. 29th Symposium on Distributed Computing (DISC'15)*, Springer LNCS 9363, pp. 420-435, 2015. (With A. Castañeda and S. Rajsbaum, UNAM, Mexico).

[C316] Eventual leader election despite crash-recovery and omission failures. *21th IEEE Int'l Pacific Rim Symposium on Dependable Computing (PRDC'15)*. IEEE Press, pp. 209-214, 2015. (With Fernńdez-Campusano F., Larrea M., Cortiñas R. –University of the Basque Country, San Sebastian, Spain–).

[C317] **Anonymous obstruction-free (n, k)-set agreement with $(n - k + 1)$ atomic read/write registers**. *Proc. 19th Int'l Conference On Principles Of Distributed Systems (OPODIS'15)*, Leibniz Int'l Proceedings in Informatics, LIPICS 46, Article 18, 17 pages (2015). (With Z. Bouzid and P. Sutra.)

[C318] Signature-free communication and agreement in the presence of Byzantine processes. *Proc. 19th Int'l Conference On Principles Of Distributed Systems (OPODIS'15)*, Leibniz Int'l Proceedings in Informatics, LIPICS 46, Article 1, 11 pages (ISBN 978-3-939897-98-9) (2015).

[C319] A Communication-efficient leader election algorithm in partially synchronous systems prone to crash-recovery and omission failures. *Proc. 17th Int'l Conference on Distributed Computing and Networking (ICDCN'16)*, ACM Press, Article 5, 6 pages, 2016. (With M. Larea, Ch. Fernãndez, and R. Cortiñas –University of the Basque Country, San Sebastian, Spain–.)

[C320] Modular randomized Byzantine k-set agreement in asynchronous message-passing systems. *Proc. 17th Int'l Conference on Distributed Computing and Networking (ICDCN'16)*, ACM Press, Article 8, 10 pages, 2016. (With Achour Mostéfaoui and Hamouma Moumen.)

[C321] Efficient broadcast protocol for Internet of things. (160/541.) (With H. Lakhlef and J. Bourgeois). *30th IEEE Int'l Conference on Advanced Information Networking and Applications (AINA'16)*. IEEE Computer Press, pp. 999-1005, March 2016.

[C322] A look at basics of distributed computing. Invited tutorial. *Proc. 36th IEEE Int'l Conference on Distributed Computing (ICDCS'16)*, IEEE Press, pp. 1-11, 2016.

[C323] Time-efficient atomic read/write register in crash-prone asynchronous message-passing systems. (21/124). *Proc. 4th Int'l Conference on Networked Systems (NETYS'16)*, Springer LNCS 9944, pp. 250-265, 2016. (With Achour Mostéfaoui.)

[C324] **Two-bit messages are sufficient to implement atomic read/write registers in crash-prone systems**. (40/137) *Proc. 35th ACM Symposium on Principles of Distributed Computing (PODC'16)*, ACM Press, pp. 381-390, 2016. (With Achour Mostéfaoui.)

[C325] Optimal collision/conflict-free distance-2 coloring in wireless broadcast/receive synchronous tree networks. (54/251). *Proc. 45th Annual Conference on Parallel Processing (ICPP'16)*, pp. 350-359, 2016. (With Davide Frey and Hicham Lakhlef.)

[C326] **t-Resilient immediate snapshot is impossible**. *Proc. 23nd Int'l Colloquium on Structural Information and Communication Complexity (SIROCCO'16)*, Springer LNCS 9988, pp. 177-191, 2016. (With C. Delporte and H. Fauconnier, Université Paris 7–, and S. Rajsbaum –UNAM, Mexico–.)

[C327] **Are Byzantine failures really different from crash failures?** (32/132). *Proc. 30th Symposium on Distributed Computing (DISC'16)*, Springer LNCS 9888, pp. 215-229, 2016. (With Damien Imbs and Julien Stainer.)

[C328] Vertex coloring with communication and local memory constraints in synchronous broadcast networks. *Proc. 12th Int'l Symposium on Algorithms and Experiments for Wireless Sensor Networks (ALGOSENSORS'16)*, Springer LNCS 10050, pp. 29-44, 2016. (With Hicham Lakhlef and François Taïani.)

[C329] Implementing snapshot objects on top of crash-prone asynchronous message-passing systems. *Proc. 16th Int'l Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'16)*, Springer LNCS 10048, pp. 341-355, 2016. (With C. Delporte and H. Fauconnier –Paris 7–, and S. Rajsbaum –UNAM, Mexico–.)

[C330] **Making local algorithms wait-free, the case of ring coloring**. *Proc. 18th Int'l Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'16)*, Springer LNCS 10083, pp. 109-125, 2016. (With A. Castañeda and S. Rajsbaum, UNAM, Mexico, and C. Delporte and H. Fauconnier, –IRIF, Paris 7–.)

[C331] Providing collision-free and conflict-free communication in general synchronous broadcast/receive networks. *Proc.31th IEEE Int'l Conference on Advanced Information Networking and Applications (AINA'17)*. IEEE Computer Press, pp. 309-406 (2017). (With A. Bouhabdallah, Hicham Lakhlef and François Taïani.)

[C332] Early decision in synchronous consensus: a predicate-based guided tour. *Proc. 5th Int'l Conference on Networked Systems (NETYS'17)*, Springer LNCS 10299, pp. 206-221, 2017. (With A. Castañeda, Y. Moses, and M. Roy.)

[C333] **Long-lived tasks**. *Proc. 5th Int'l Conference on Networked Systems (NETYS'17)*, Springer LNCS 10299, pp. 439-454, 2017. (With A. Castañeda and S. Rajsbaum.)

[C334] **Which broadcast abstraction captures k-set agreement?** *Proc. 31th Int'l Symposium on Distributed Computing (DISC'17)*, Leibniz Int'l Proceedings in Informatics LIPICs, Vol. 91, Article 27, 16 pages (2017) (With D. Imbs, A. Mostéfaoui, and M. Perrin).

[C335] **Set-constrained delivery broadcast: definition, abstraction power, and computability limits**. *Proc. 19th Int'l Conference on Distributed Computing and Networking (ICDCN'18)*, ACM Press, Article 7, 10 pages, 2018. (With D. Imbs, A. Mostéfaoui, and M. Perrin).

[C336] Anonymity in distributed read/write systems: a short introduction. *Proc. 6th Int'l Conference on Networked Systems (NETYS'18)*, Springer LNCS 11028, pp. 122-140 (2018) (With Jiannong Cao, Polytechnic University, Hong Kong).

[C337] Time-efficient RFID-based stocktaking with a coarse-grained inventory list. *Proc. IEEE/ACM International Symposium on Quality of Service (IWQoS'18)*. IEEE/ACM Press, 6 pages (2018) (With Weiping Zhu Xing Meng, Xiaolei Peng –Wuhan University, RPC– and Jiannong Cao –Polytechnic University Hong Kong–).

[C338] A pleasant stroll through the land of distributed machines, computation, and universality. Invited talk. *Proc. 8th In'l Conference on Machines, Computations, and Universality (MCU'18)*, Springer LNCS 10881, pp. 34-50 (2018). (With Jiannong Cao, Polytechnic Univesrity, Hong Kong).

[C339] Set agreement and renaming in the presence of contention-related crash failures. *Proc. 20th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'18)*, Springer LNCS 11201, pp. 269-283 (2018). (With A. Durand and G. Taubenfeld).

[C340] Bee's strategy against byzantines: replacing Byzantine participants. *Proc. 20th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'18)*, Springer LNCS 11201, pp. 139-153 (2018). (With R. Baldoni and S. Bonomi –La Sapienzia, Roma– and S. Dolev and A. Shaer –Ben Gourion University, Beer Sheva–).

[C341] DBFT: efficient Byzantine consensus with a weak coordinator and its application to consortium blockchains. *17th IEEE International Symposium on Network Computing and Applications (NCA'18)*, IEEE Press (2018) (With T. Crain and V. Gramoli –Sydney University–, M. Larrea –Basque Country University–).

[C342] On the weakest failure detector for read/write-based mutual exclusion. *Proc. 33nd Int'l Conference on Advanced Information Networking and Applications (AINA'19)*. Springer Series "Advances in Intelligent Systems and Computing", Vol. AISC 926, pp. 272-285 (2019). (With C. Delporte and H. Fauconnier, IRIF, Paris 7).

[C343] **One for all and all for one: scalable consensus in a hybrid communication model.** *Proc. 39th IEEE Int'l Conference on Distributed Computing (ICDCS'19)*, IEEE Press, pp. 464-471, (2019) (With J. Cao, Hong Kong Polytechnic University).

[C344] Mutex-based de-anonymization of an anonymous read/write memory. *Proc. 6th Int'l Conference on Networked Systems (NETYS'19)*, Springer LNCS 11704, pp. 311-326 (2019) (With E. Godard and D. Imbs, LIS, Université d'Aix- Marseille, and G. Taubenfeld, Interdisciplinary Center, Herzliya, Israël).

[C345] **Optimal memory-anonymous symmetric deadlock-free mutual exclusion**. *Proc. 38th ACM Symposium on Principles of Distributed Computing (PODC'19)*, ACM press, pp. 157-166 (2019) (With Z. Aghazadeh and Ph. Woelfel –University of Calgary, Canada–, D. Imbs –Université d'Aix-Marseille–, and G. Taubenfeld –Israël–).

[C346] Participant-restricted consensus in asynchronous crash-prone read/write systems and its weakest failure detector. *Proc. 15th International Conference on Parallel Computing Technologies (PaCT'19)*, Springer, LNCS 11657, pp. 419-430 (2019) (With C. Delporte and H. Fauconnier, IRIF, Paris 7).

[C347] Anonymous read/write memory: leader election and de-anonymization. *Proc. 26th Int'l Colloquium on Structural Information and Communication Complexity (SIROCCO'19)*, Springer LNCS 11639, pp. 246-261 (2019) (With E. Godard and D. Imbs –LIS, Université d'Aix- Marseille– and G. Taubenfeld –Interdisciplinary Center, Herzliya, Israël–).

[C348] Fully anonymous shared memory algorithms. *Proc. 21th Int'l Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2019)*, Springer LNCS 11914, pp. 301-306 (2018) (With G. Taubenfeld.)

[C349] A new insight into local coin-based randomized consensus. *Proc. 24th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'19)*, IEEE Press, pp. 207-216, (2019) (With A. Mostéfaoui and M. Perrin –Université de Nantes, France–).

[C350] **The Notion of universality in crash-prone asynchronous message-passing systems: a tutorial.** Invited Tutorial. *Proc. 38th International Symposium on Reliable Distributed Systems (SRDS'19)*, IEEE Computer Press, pp. 334-350 (2019)

[C351] Byzantine-tolerant set-constrained delivery (SCD) broadcast. *Proc. 23th Int'l Conference On Principles Of Distributed Systems (OPODIS'19)*, Leibniz Int'l Proceedings in Informatics, LIPICS Vol. 153, Article 6, 23 pages (2019). (With A. Auvolat and F. Taïani.)

[C352] Fully anonymous consensus and set agreement algorithms. *Proc. 8th Int'l Conference on Networked Systems (NETYS'20)*, Springer LNCS 12129, pp. 314-328 (2020) (With G. Taubenfeld, Interdisciplinary Center, Herzliya, Israel)

[C353] Self-stabilizing uniform reliable broadcast. *8th Int'l Conference on Networked Systems (NETYS'20)*, Springer LNCS 12129, pp. 296-313 (2020) (With O. Lundström, Elad M. Schiller -Chalmers University, Sweden-).

[C354] **From Bezout identity to space-optimal leader election in anonymous memory systems**. *Proc. 39th ACM Symposium on Principles of Distributed Computing (PODC'20)*, ACM press, pp. 41-50 (2020) (With E. Godard and D. Imbs –LIS, Université d'Aix-Marseille– and G. Taubenfeld –Interdisciplinary Center, Herzliya, Israel–.)

[C355] Self-stabilizing set-constrained delivery broadcast (105/584). *40th IEEE Int'l Conference on Distributed Computing Systems (ICDCS'20)*, IEEE Press, (2020) (With Oskar Lundström and Elad M. Schiller –Chalmers University, Sweden–).

[C356] Atomic appends in asynchronous Byzantine distributed ledgers. *Proc. 16th European Dependable Computing Conference (EDCC'20)*, IEEE Publisher, pp. 77-84 (2020). (With V. Cholvi, Fernández Anta A. –Spain–, Georgiou Ch., and Nicolaou N. – Cyprus–.)

[C357] **k-Immediate snapshot and x-set agreement: how are they related?** *Proc. 22th Int'l Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'20)* Springer LNCS 12514, pp. 97-112 (2020) (With C. Delporte and H. Fauconnier –Université Paris–, and S. Rajsbaum –UNAM, Mexico–.)

[C358] Eventual leader election in the ADD communication model. *Proc. 22th Int'l Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'20)* Springer LNCS 12514, pp. 229-234 (2020) (With K. Vargas and S. Rajsbaum –UNAM, Mexico–).

[C359] **Relaxed queues and stacks from read/write operations.** *Proc.24th Conference on Principles of Distributed Systems (OPODIS 2020)*, LIPICS Vol. 184, Article 13, 19 pages (2020). (With A. Castañeda and and S. Rajsbaum –UNAM, Mexico–).

[C360] Self-stabilizing indulgent zero-degrading binary consensus. *Proc. 22th Int'l Conference on Distributed Computing and Networking (ICDCN'21)*, ACM Press, pp. 106-115 (2021) (With O. Lundström and E. M. Schiller –Chalmers University, Sweden–).

[C361] From incomplete to complete networks in asynchronous Byzantine systems. To appear *35th Int'l Conference on Advanced Information Networking and Applications (AINA'21)* (2021) (With J. Cao, HK Polytechnic Univerity).

——————————— Current submissions to conferences ———————————

[C362] Leader election in arbitrarily connected networks with process crashes and weak channel reliability. (With C. López, K. Vargas and S. Rajsbaum –UNAM, Mexico–).