RESEARCH CENTRE

Rennes - Bretagne Atlantique

IN PARTNERSHIP WITH:

Université Rennes 1

2020 ACTIVITY REPORT

Project-Team WIDE

the World Is Distributed Exploring the tension between scale and coordination

IN COLLABORATION WITH: Institut de recherche en informatique et systèmes aléatoires (IRISA)

DOMAIN

Networks, Systems and Services, Distributed Computing

THEME

Distributed Systems and middleware

Contents

Project-Team WIDE

Creation of the Team: 2018 January 01, updated into Project-Team: 2018 June 01

Keywords

Computer sciences and digital sciences

A1.2.5. – Internet of things
A1.2.9. – Social Networks
A1.3.2. – Mobile distributed systems
A1.3.3. – Blockchain
A1.3.4. – Peer to peer
A1.3.5. – Cloud
A1.3.6. – Fog, Edge
A2.1.7. – Distributed programming
A2.6.2. – Middleware
A2.6.3. – Virtual machines
A3.5.1. – Analysis of large graphs
A4. – Security and privacy
A4.8. – Privacy-enhancing technologies
A7.1.1. – Distributed algorithms
A7.1.2. – Parallel algorithms
A7.1.3. – Graph algorithms
A9. – Artificial intelligence
A9.2. – Machine learning
A9.9. – Distributed AI, Multi-agent

Other research topics and application domains

B6.1.1. – Software engineering
B6.3.1. – Web
B6.3.5. – Search engines
B6.4. – Internet of things
B9.5.1. – Computer science
B9.5.6. – Data science

1 Team members, visitors, external collaborators

Research Scientists

- Davide Frey [Inria, Researcher, HDR]
- George Giakkoupis [Inria, Researcher]
- Erwan Le Merrer [Inria, Advanced Research Position, HDR]

Faculty Members

- Francois Taiani [Team leader, Univ de Rennes I, Professor, HDR]
- David Bromberg [Univ de Rennes I, Professor, HDR]
- Michel Raynal [Univ de Rennes I, Emeritus, HDR]

Post-Doctoral Fellow

- Noureddine Haouari [Univ de Rennes I, until Oct 2020]

PhD Students

- Alex Auvolat-Bernstein [Univ de Rennes I]
- Loick Bonniot [Technicolor, CIFRE]
- Amaury Bouchra Pilet [Univ de Rennes I]
- Quentin Dufour [Inria]
- Louison Gitzinger [Univ de Rennes I, until Sep 2020]
- Florestan de Moor [École normale supérieure de Rennes, until Aug 2020]

Technical Staff

- Jeremie Dautheribes [Inria, Engineer, from Dec 2020]
- Malo Dumont [Inria, Engineer, from Jun 2020]
- Matthieu Simonin [Inria, Engineer, from Apr 2020]

Interns and Apprentices

- Jad Alhajj [Univ de Rennes I, from May 2020 until Aug 2020]
- Paul Bastide [École normale supérieure de Rennes, from Jun 2020 until Aug 2020]
- Mathieu Gestin [Univ de Rennes I, from Sep 2020]
- Josselin Giet [École Normale Supérieure de Paris, until Jan 2020]
- Lucie Guillou [Univ de Rennes I, from May 2020 until Jul 2020]
- Guillaume Longrais [Univ de Rennes I, from May 2020 until Aug 2020]
- Tomas Martinez [Inria, until Mar 2020]
- Michael Paper [Inria, from Jun 2020 until Jul 2020]
- Robinson Sablons De Gelis [Univ de Rennes I, from May 2020 until Jul 2020]
- Emmanuel Tran [Univ de Rennes I, from Mar 2020 until Aug 2020]

Administrative Assistant

- Virginie Desroches [Inria]

2 Overall objectives

2.1 Overview

The long term goal of the WIDE team is to provide the practical tools and theoretical foundations required to address the scale, dynamicity, and uncertainty that constitute the foundations of modern distributed computer systems. In particular, we would like to **explore the inherent tension between scalability and coordination guarantees**, and develop novel techniques and paradigms that are adapted to the rapid and profound changes impacting today's distributed systems, both in terms of the application domains they support and the operational constraints they must meet.

These changes are particularly visible in three key areas related to our research: *(i)* planetary-scale information systems, *(ii)* personalized services, and *(iii)* new forms of social applications (e.g. in the field of the sharing economy).

2.2 Planetary-Scale Geo-Distributed Systems

Modern large-scale systems often encompass thousands of server nodes, hosted in tens of datacenters distributed over several continents. To address the challenges posed by such systems, alternative distributed architectures are today emerging that emphasize *decentralized* and *loosely coupled* interactions. This evolution can be observed at multiple levels of an application's distributed stack: the growing interest, both practical and theoretical, for weak consistency models is such an example. In spite of their potential counter-intuitive behaviors, weakly consistent data-structures allow developers to trade strict coordination guarantees for the ability to deliver a reactive and scalable service even when hit by arbitrary network delays or system partitions. At a higher, more architectural level, similar motivations explain the push for *micro-services* on the server side of on-line applications and the growth of rich *browser-based programming technologies* on their client side. Micro services help development teams decompose complex applications into a set of simpler and loosely-connected distributed services. In a parallel evolution, modern browsers embark increasingly powerful networking APIs such as WebRTC. These APIs are prompting a fresh rethink of the typical distribution of capabilities between servers and clients. This is likely to lead to more services and computations being offloaded to browsers, in particular within hybrid architectures. The above evolutions, away from tightly synchronized and monolithic deployments towards heterogeneous, composite and loosely coordinated distributed systems, raise a number of difficult challenges at the crossroad of theoretical distributed algorithms, system architecture, and programming frameworks. One of these challenges pertains to the growing complexity arising from these systems: as richer and more diverse services are being composed to construct whole applications, individual developers can only hope to grasp parts of the resulting systems. Similarly, weak consistency models and loose coordination mechanisms tend to lead to counter-intuitive behaviors, while only providing weak overall guarantees. This lack of systematic guarantees and understandability make it harder for practitioners to design, deploy, and validate the distributed systems they produce, leading to rising costs and high entry barriers.

In order to address these challenges, we argue that modern-day distributed systems require new principled algorithms, approaches, and architectural patterns able to provide sound foundations to their development while guaranteeing robust service guarantees, thus lowering the cost of their development and maintenance, increasing their reliability, and rendering them technically approachable to a wider audience.

2.3 Highly Personalized On-Line Services

Ever increasing volumes of data are being produced and made available from a growing number of sources (Internet of Things sensors, open data repositories, user-generated content services).

As a result, digital users find it increasingly difficult to face the data deluge they are subjected to without additional help. This difficulty has fueled the rise of notification solutions over traditional search, in order to push few but relevant information items to users rather than leave them to sieve through a large mass of non-curated data. To provide such personalized services, most companies rely today on centralized or tightly coupled systems hosted in data centers or in the cloud. These systems use advanced data-mining and machine learning techniques to deliver enhanced, personalized, services to users and companies, and often exploit highly parallelized data analytics frameworks such as Spark, and Flink.

Selecting the best information for a user in order to provide a personalized experience requires however to gather enough information about this user, which raises a number of important technical challenges and privacy protection issues. More precisely, this concentration poses strong risks to the privacy of users, and limits the scope of personalization to tightly integrated datasets.

The use of large monolithic infrastructures also limits the use of machine learning and personalization to situations in which data is fully available to the organization managing the underlying computing infrastructure. This set-up prevents for instance cases in which sensitive data may not be shared freely, but might be of mutual interest to several independent participants in order to construct common machine learning models usable by all. Such situations occur for instance in the context of the mining of health-records by independent health-organizations, or in the collective harnessing of individual on-line profiles for personalization purpose by private users.

Alternative decentralized approaches that eschew the need for a central all-encompassing authority holds the promise of delivering knowledge while protecting individual participants. Constructing such systems requires however to address the inherent tension between the need to limit sensitive individual leaks, while maximizing collectively gained insights. Answering this tension calls on techniques and approaches from distributed systems, information theory, security, and randomized processes, making it a rich and dense research area, with a high impact potential. The problem of distributed privacy in a digital interconnected age further touches on interdisciplinary questions of Law, Sociology and Public Policy, which we think can only be explored in collaboration with colleagues from these fields.

2.4 Social Collaboration Platforms

On-line social networks have had a fundamental and lasting impact on the Internet. In recent years, numerous applications have appeared that go beyond the services originally provided by “pure” on-line social networks, such as posting messages or maintaining on-line “friendship” links. These new applications seek to organize and coordinate users, often in the context of the sharing economy, for instance in order to facilitate car-sharing (e.g. BlaBla car, https://www.blablacar.com/), short-term renting (e.g. AirBnB, https://www.airbnb.com/), and peer-to-peer financial services (e.g. Lending Club, https://www.lendingclub.com/). Some systems, such as Bitcoin or Ethereum, have given rise to new distributed protocols combining elements of cryptography and distribution that are now largely discussed in the research community, and have attracted the attention of policy makers and leading financial actors.

The challenges faced by such social applications blend in many ways issues already discussed in the two previous subsections and cast them in an application-driven context. These social collaboration platforms require mechanisms that go beyond pure message propagation, with stricter consistency and robustness guarantees. Because they involve connected users, these applications must provide usable solutions, in particular in terms of latency and availability. At the same time, because they manipulate real-world transactions and objects (money, cars, accommodations) they must also provide a high level of consistency and guarantees. Many of these applications further operate at a planetary scale, and therefore also face stark scalability issues, that make them highly interesting case studies to investigate innovative architectures combining decentralized and centralized elements.

Formalizing and characterizing the needs and behaviors of these new applications seems particularly interesting in order to provide the fertile ground for new systems and novel theoretical work. The area of social applications also offers avenues for knowledge transfer and societal impact, along two dimensions. First, practical and usable approaches, back by a deep understanding of the foundation of distribution and coordination, are likely to find applications in future systems. Second, developers of complex social applications are often faced with a lack of robust scalable services [1] that can be easily exploited to harness

[1] The repeated debugging of MongoDB’s replication algorithm (e.g. see https://aphyr.com/posts/338-jepsen-mongodb

the latest understanding of large-scale distributed coordination. We therefore think these applications offer an opportunity to design and deliver modular reusable bricks that can be easily appropriated by a large population of innovative developers without requiring the level of deep understanding usually necessary to implement these solutions from scratch. Providing such reusable bricks is however difficult, as many interesting formal properties are not composable, and a unified composable theory of distributed systems still need to be fully articulated.

3 Research program

3.1 Overview

In order to progress in the four fields described above, the WIDE team is developing a research program which aims to **help developers control and master the inherent uncertainties and performance challenges brought by scale and distribution**.

More specifically, our program revolves around four key challenges.

- Challenge 1: Designing Hybrid Scalable Architectures,
- Challenge 2: Constructing Personalizable Privacy-Aware Distributed Systems,
- Challenge 3: Understanding Controllable Network Diffusion Processes,
- Challenge 4: Systemizing Modular Distributed Computability and Efficiency.

These four challenges have in common **the inherent tension between coordination and scalability in large-scale distributed systems**: strong coordination mechanisms can deliver strong guarantees (in terms of consistency, agreement, fault-tolerance, and privacy protection), but are generally extremely costly and inherently non-scalable if applied indiscriminately. By contrast, highly scalable coordination approaches (such as epidemic protocols, eventual consistency, or self-organizing overlays) perform much better when the size of a system increases, but do not, in most cases, provide any strong guarantees in terms of consistency or agreement.

The above four challenges explore these tensions from *four complementary angles*: from an architectural perspective (Challenge 1), from the point of view of a fundamental system-wide guarantee (privacy protection, Challenge 2), looking at one universal scalable mechanism (network diffusion, Challenge 3), and considering the interplay between modularity and computability in large-scale systems (Challenge 4). These four challenges range from practical concerns (Challenges 1 and 2) to more theoretical questions (Challenrges 3 and 4), yet present *strong synergies* and *fertile interaction points*. E.g. better understanding network diffusion (Challenge 3) is a key enabler to develop more private decentralized systems (Challenge 2), while the development of a theoretically sound modular computability hierarchy (Challenge 4) has a direct impact on our work on hybrid architectures (Challenge 1).

3.2 Hybrid Scalable Architectures

The rise of planetary-scale distributed systems calls for novel software and system architectures that can support user-facing applications while scaling to large numbers of devices, and leveraging established and emerging technologies. The members of WIDE are particularly well positioned to explore this avenue of research thanks to their experience on de-concentrated architectures combining principles from both decentralized peer-to-peer [54, 64] systems and hybrid infrastructures (i.e. architectures that combines centralized or hierarchical elements, often hosted in well-provisioned data-centers, and a decentralized part, often hosted in a peer-to-peer overlay) [58]. In the short term, we aim to explore two axes in this direction: browser-based communication, and micro services.

`-3-4-0-rc3`) is a telling illustration of the difficulties encountered by development teams when building such platforms.

Browser-based fog computing The dramatic increase in the amount of data being produced and processed by connected devices has led to paradigms that seek to decentralize the traditional cloud model. In 2011 Cisco [55] introduced the vision of *fog computing* that combines the cloud with resources located at the edge of the network and in between. More generally, the term *edge computing* has been associated with the idea of adding edge-of-the network storage and computation to traditional cloud infrastructures [50].

A number of efforts in this directions focus on specific hardware, e.g. fog nodes that are responsible for connected IoT devices [56]. However, many of today's applications run within web browsers or mobile phones. In this context, the recent introduction of the WebRTC API, makes it possible for browsers and smartphones to exchange directly between each other, enabling mobile, or browser-based decentralized applications.

Maygh [79], for example, uses the WebRTC API to build a decentralized Content Delivery Network that runs solely on web browsers. The fact that the application is hosted completely on a web server and downloaded with enabled websites means that webmasters can adopt the Content Delivery Network (CDN) without requiring users to install any specific software.

For us, the ability of browsers to communicate with each other using the WebRTC paradigm provides a novel playground for new programming models, and for a *browser-based fog architecture* combining both a centralized, cloud-based part, and a decentralized, browser-supported part.

This model offers tremendous potential by making edge-of-the-network resources available through the interconnection of web-browsers, and offers new opportunities for the protection of the personal data of end users. But consistently engineering browser-based components requires novel tools and methodologies.

In particular, WebRTC was primarily designed for exchanging media and data between two browsers in the presence of a coordinating server. Its complex mechanisms for connection establishment make many of the existing peer-to-peer protocols inefficient. To address this challenge, we plan to consider two angles of attack. First, we plan to design novel protocols that take into account the specific requirements set by this new technology. Second, we envisage to investigate variants of the current WebRTC model with cheaper connection-establishment protocols, in order to provide lower delays and bandwidth consumption in large-scale browser-based applications.

We also plan to address the trade-offs associated with hybrid browser-cloud models. For example, when should computation be delegated to browsers and when should it be executed on the cloud in order to maximize the quality of service? Or, how can a decentralized analytics algorithms operating on browser-based data complement or exploit the knowledge built by cloud-based data analytics solutions?

Emergent micro-service deployment and management Micro-services tend to produce fine-grained applications in which many small services interact in a loosely coupled manner to produce a wide range of services within an organization. Individual services need to evolve independently of each other over time without compromising the availability of the overall application. Lightweight isolation solutions such as containers (Docker, ...), and their associated tooling ecosystem (e.g. Google's Borg [78], Kubernetes [53]) have emerged to facilitate the deployment of large-scale micro-service-based applications, but only provide preliminary solutions for key concerns in these systems, which we would like to investigate and extend.

Most of today's on-line computer systems are now too large to evolve in monolithic, entirely pre-planned ways. This applies to very large data centres, for example, where the placement of virtual machines to reduce heating and power consumption can no longer be treated using top-down exhaustive optimisation approaches beyond a critical size. This is also true of social networking applications, where different mechanisms—e.g. to spread news notifications, or to recommend new contacts—must be adapted to the different sub-communities present in the system.

To cope with the inherent complexity of building complex loosely-coupled distributed systems while fostering and increasing efficiency, maintainability, and scalability, we plan to study how novel programming techniques based on declarative programming, components and epidemic protocols can help design, deploy, and maintain self-adaptive structures (e.g. placement of VM) and mechanisms (e.g. contact recommendations) that are optimized to the local context of very large distributed systems. To fulfill this vision, we plan to explore a three-pronged strategy to raise the level of programming abstraction offered to developers.

- First, we plan to explore the use of high-level domain-specific languages (DSL) to declare how large-scale topologies should be achieved, deployed, and maintained. Our vision is a declarative approach to describe how to combine, deploy and orchestrate micro-services in an abstract manner thus abstracting away developers from the underlying cloud infrastructures, and from the intricacies involved in writing low-level code to build a large-scale distributed application that scales. With this effort, we plan notably to directly support the twin properties of *emergence* (the adaptation "from within") and *differentiation* (the possibility from parts of the system to diverge while still forming a whole). Our central objective is to search for principled programming constructs to support these two capabilities using a modular and incremental software development approach.

- On a second strand of work, we plan to investigate how unikernels enable smaller footprints, more optimization options, and faster boot times for micro-services. Isolating micro-services into VMs is not the most adequate approach as it requires the use of hypervisors, or virtual machine monitors (VMMs), to virtualize hardware resources. VMMs are well known to be heavyweight with both boot and run time overheads that may have a strong impact on performances. Unikernels seem to offer the right balance between performance and flexibility to address this challenge. One of the key underlying challenges is to compile directly the aforementioned provided DSL to a dedicated and customized machine image, ready to be deployed directly on top of a large set of bare metal servers.

- Depending on the workload it is subjected to, and the state of its execution environment (network, VMs), a large-scale distributed application may present erratic or degraded performance that is hard to anticipate and plan for. There is therefore a strong need to adapt dynamically the way resources are allocated to a running application. We would like to study how the DSL approach we envisage can be extended to enable developers to express orchestration algorithms based on machine learning algorithms.

3.3 Personalizable Privacy-Aware Distributed Systems

On-line services are increasingly moving towards an in-depth analysis of user data, with the objective of providing ever better personalization. But in doing so, personalized on-line services inevitably pose risks to the privacy of users. Eliminating, or even reducing these risks raises important challenges caused by the inherent trade-off between the level of personalization users wish to achieve, and the amount of information they are willing to reveal about themselves (explicitly or through the many implicit sources of digital information such as smart homes, smart cars, and IoT environments).

At a general level, we would like to address these challenges through protocols that can provide access to unprecedented amounts of data coming from sensors, users, and documents published by users, while protecting the privacy of individuals and data sources. To this end, we plan to rely on our experience in the context of distributed systems, recommender systems, and privacy, as well as in our collaborations with experts in neighboring fields such as machine learning, and security. In particular, we aim to explore different privacy-utility tradeoffs that make it possible to provide differentiated levels of privacy guarantees depending on the context associated with data, on the users that provide the data, and on those that access it. Our research targets the general goal of privacy-preserving decentralized learning, with applications in different contexts such as user-oriented applications, and the Internet-of-Things (IoT).

Privacy-preserving decentralized learning Personalization and recommendation can be seen as a specific case of general machine learning. Production-grade recommenders and personalizers typically centralize and process the available data in one location (a data-center, a cloud service). This is highly problematic, as it endangers the privacy of users, while hampering the analysis of datasets subject to privacy constraints that are held by multiple independent organizations (such as health records). A decentralized approach to machine learning appears as a promising candidate to overcome these weaknesses: if each user or participating organization keeps its data, while only exchanging gradient or model information, privacy leaks seem less likely to occur.

In some cases, decentralized learning may be achieved through relatively simple adaptations of existing centralized models, for instance by defining alternative learning models that may be more easily

decentralized. But in all cases, processing growing amounts of information calls for high-performance algorithms and middleware that can handle diverse storage and computation resources, in the presence of dynamic and privacy-sensitive data. To reach this objective, we will therefore leverage our work in distributed and privacy-preserving algorithms and middleware [57, 59, 60] as well as the results of our work on large-scale hybrid architectures in Objective 1.

Personalization in user-oriented applications As a first application perspective, we plan to design tools that exploit decentralized analytics to enhance user-centric personalized applications. As we observed above, such applications exhibit an inherent trade-off between personalization quality and privacy preservation. The most obvious goal in this direction consists in designing algorithms that can achieve high levels of personalization while protecting sensitive user information. But an equally important one consists in personalizing the trade-off itself by adapting the quality of the personalization provided to a user to his/her willingness to expose information. This, like other desirable behaviors, appears at odds with the way current systems work. For example, a user of a recommender system that does not reveal his/her profile information penalizes other users causing them to receive less accurate recommendations. We would like to mitigate this situation by means of protocols that reward users for sharing information. On the one hand, we plan to take inspiration from protocols for free-riding avoidance in peer-to-peer systems [61, 66]. On the other hand, we will consider blockchains as a tool for tracking and rewarding data contributions. Ultimately, we aim at enabling users to configure the level of privacy and personalization they wish to experience.

Privacy preserving decentralized aggregation As a second setting we would like to consider target applications running on constrained devices like in the Internet-of-Things (IoT). This setting makes it particularly important to operate on decentralized data in a light-weight privacy-preserving manner, and further highlights the synergy between this objective and Objective 1. For example, we plan to provide data subjects with the possibility to store and manage their data locally on their own devices, without having to rely on third-party managers or aggregators, but possibly storing less private information or results in the cloud. Using this strategy, we intend to design protocols that enable users themselves, or third-party companies to query distributed data in aggregate form, or to run data analytics processes on a distributed set of data repositories, thereby gathering knowledge without violating the privacy of other users. For example, we have started working on the problem of computing an aggregate function over a subset of the data in a distributed setting. This involves two major steps: selection and aggregation. With respect to selection, we envision defining a decentralized data-selection operation that can apply a selection predicate without violating privacy constraints. With respect to aggregation, we will continue our investigation of lightweight protocols that can provide privacy with limited computational complexity [51].

3.4 Network Diffusion Processes

Social, biological, and technological networks can serve as conduits for the spread of ideas, trends, diseases, or viruses. In social networks, rumors, trends and behaviors, or the adoption of new products, spread from person to person. In biological networks, diseases spread through contact between individuals, and mutations spread from an individual to its offsprings. In technological networks, such as the Internet and the power grid, viruses and worms spread from computer to computer, and power failures often lead to cascading failures. The common theme in all the examples above is that the rumor, disease, or failure starts out with a single or a few individual nodes, and propagates through the network, from node to node, to reach a potentially much larger number of nodes.

These types of *network diffusion processes* have long been a topic of study in various disciplines, including sociology, biology, physics, mathematics, and more recently, computer science. A main goal has been to devise mathematical models for these processes, describing how the state of an individual node can change as a function of the state of its neighbors in the network, and then analyse the role of the network structure in the outcome of the process. Based on our previous work, we would like to study to what extent one can affect the outcome of the diffusion process by controlling a small, possibly carefully selected fraction of the network.

For example, we plan to explore how we may increase the spread or speed of diffusion by choosing an appropriate set of seed nodes (a standard goal in viral marketing by word-of-mouth), or achieve the opposite effect either by choosing a small set of nodes to remove (a goal in immunization against diseases), or by seeding a competing diffusion (e.g., to limit the spread of misinformation in a social network).

Our goal is to provide a framework for a systematic and rigorous study of these problems. We will consider several standard diffusion models and extensions of them, including models from mathematical sociology, mathematical epidemiology, and interacting particle systems. We will consider existing and new variants of spread maximization/limitation problems, and will provide (approximation) algorithms or show negative (inapproximability) results. In case of negative results, we will investigate general conditions that make the problem tractable. We will consider both general network topologies and specific network models, and will relate the efficiency of solutions to structural properties of the topology. Finally, we will use these insights to engineer new network diffusion processes for efficient data dissemination.

Spread maximization Our goal is in particular to study spread maximization in a broader class of diffusion processes than the basic independent cascade (IC) and linear threshold (LT) models of influence [73, 71, 72] that have been studied in this context so far. This includes the *randomized rumor spreading (RS)* model for information dissemination [63], *biased* versions of the *voter model* [68] modelling influence, and the (graph-based) *Moran processes* [75] modelling the spread of mutations. We would like to consider several natural versions of the spread maximization problem, and the relationships between them. For these problems we will use the greedy algorithm and the submodularity-based analytical framework of [73], and will also explore new approaches.

Immunization optimization Conversely we would also like to explore immunization optimization problems. Existing works on these types of problem assume a *perfect-contagion* model, i.e., once a node gets infected, it deterministically infects all its non-immunized neighbors. We plan to consider various diffusion processes, including the standard *susceptible–infected* (SI), *susceptible–infected–recovered* (SIR) and *susceptible–infected–susceptible* (SIS) epidemic models, and explore the extent to which results and techniques for the perfect-contagion model carry over to these probabilistic models. We will also investigate whether techniques for spread maximization could be applied to immunization problems.

Some immunization problems are known to be hard to approximate in general graphs, even for the perfect-contagion model, e.g., the fixed-budget version of the fire-fighter problem cannot be approximated to any $n^{1-\epsilon}$ factor [52]. This strand of work will consider restricted graph families, such as trees or graphs of small treewidth, for such problems. In addition, for some immunization problems, there is a large gap between the best known approximation algorithm and the best known inaproximability result, and we would like to make progress in reducing these gaps.

3.5 Systemizing Modular Distributed Computability and Efficiency

The applications and services envisaged in Objectives 1 and 2 will lead to increasingly complex and multifaceted systems. Constructing these novel hybrid and decentralized systems will naturally push our need to understand distributed computing beyond the current state of the art. These trends therefore demand research efforts in establishing sound theoretical foundations to allow everyday developers to master the design, properties and implementation of these systems.

We plan to investigate these foundations along two directions: first by studying novel approaches to some fundamental problems of *mutual exclusion and distributed coordination*, and second by exploring how we can build a *comprehensive and modular framework* capturing the foundations of *distributed computation*.

Randomized algorithm for mutual exclusion and coordination To exploit the power of massive distributed applications and systems (such as those envisaged in Objectives 1 and 2) or multiple processors, algorithms must cope with the scale and asynchrony of these systems, and their inherent instability, e.g., due to node, link, or processor failures. Our goal is to explore the power and limits of randomized

algorithms for large-scale networks of distributed systems, and for shared memory multi-processor systems, in effect providing fundamental building blocks to the work envisioned in Objectives 1 and 2.

For shared memory systems, randomized algorithms have notably proved extremely useful to deal with asynchrony and failures. Sometimes probabilistic algorithms provide the only solution to a problem; sometimes they are more efficient; sometimes they are simply easier to implement. We plan to devise efficient algorithms for some of the fundamental problems of shared memory computing, such as mutual exclusion, renaming, and consensus.

In particular, looking at the problem of *mutual exclusion*, it is desirable that mutual exclusion algorithms be *abortable*. This means that a process that is trying to lock the resource can abort its attempt in case it has to wait too long. Abortability is difficult to achieve for mutual exclusion algorithms. We will try to extend our algorithms for the *cache-coherent* (CC) and the *distributed shared memory* (DSM) model in order to make them abortable, while maintaining expected constant *Remote Memory References* (RMRs) complexity, under optimistic system assumptions. In order to achieve this, the algorithm will use strong synchronization primitives, called compare-and-swap objects. As part of our collaboration with the University of Calgary, we will work on implementing those objects from registers in such a way that they also allow aborts. Our goal is to build on existing non-abortable implementations [65]. We plan then later to use these objects as building blocks in our mutual exclusion algorithm, in order to make them work even if the system does not readily provide such primitives.

We have also started working on blockchains, as these represent a new and interesting trade-off between probabilistic guarantees, scalability, and system dynamics, while revisiting some of the fundamental questions and limitations of consensus in fault-prone asynchronous systems.

Modular theory 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 state machine replication). In spite of the high conceptual and practical interest of such frameworks, many of these efforts lack a sound grounding in distributed computation theory (with the notable exceptions of JGroups and Zookeeper), and often provide punctual and partial solutions for a narrow range of services. We argue that this is because we still lack a generic framework that unifies the large body of fundamental knowledge on distributed computation that has been acquired over the last 40 years.

To overcome this gap we would like to develop a systematic model of distributed computation that organizes the functionalities of a distributed computing system into reusable modular constructs assembled via well-defined mechanisms that maintain sound theoretical guarantees on the resulting system. This research vision arises from the strong belief that distributed computing is now mature enough to resolve the tension between the social needs for distributed computing systems, and the lack of a fundamentally sound and systematic way to realize these systems.

To progress on this vision, we plan in the near future to investigate, from a distributed software point of view, the impact due to failures and asynchrony on the layered architecture of distributed computing systems. A first step in this direction will address the notions of *message adversaries* (introduced a long time ago in [77]) and *process adversaries* (investigated in several papers, e.g. [76, 62, 69, 70, 74]). The aim of these notions is to consider failures, not as "bad events", but as part of the normal behavior of a system. As an example, when considering round-based algorithms, a message adversary is a daemon which, at every round, is allowed to suppress some messages. The aim is then, given a problem P, to find the strongest adversary under which P can be solved ("strongest" means here that giving more power to the adversary makes the problem impossible to solve). This work will allow us to progress in terms of general *layered* theory of distributed computing, and allow us to better *map* distributed computing models and their relations, in the steps of noticeable early efforts in this direction [76, 49].

4 Application domains

5 Highlights of the year

- Best paper award at the 2020 ACM/IFP International Conference on Middleware for [30].

- Loïck Bonniot and his team 'Steredeg' won 1st place at the Graph Neural Networking Challenge 2020, organized as part of the AI/ML in 5G Challenge of ITU (the International Telecommunication Union).
- Paper published by Nature for Erwan Le Merrer [19]
- Davide Frey and the Coronasurveys team were finalists (top 5) in the Facebook COVID-19 Symptom Data Challenge.

6 New software and platforms

6.1 New software

6.1.1 KIFF

Name: KIFF: An impressively fast and efficient JAVA library for KNN construction

Keyword: KNN

Functional Description: This package implements the KIFF algorithm reported in [1]. KIFF is a generic, fast and scalable K-Nearest-Neighbor graph construction algorithm. This algorithm connects each object to its k most similar counterparts, according to a given similarity metric. In term of comparison, this package implements also HYREC [2] and NN-DESCENT [3]. The standalone program implements cosine similarity only, however this library supports arbitrary similarity measures.

[1] Antoine Boutet, Anne-Marie Kermarrec, Nupur Mittal, Francois Taiani. Being prepared in a sparse world: the case of KNN graph construction. ICDE 2016, Finland.

Contact: Antoine Boutet

Partner: LIRIS

6.1.2 Dietcoin

Keywords: Blockchain, Bitcoin, Smartphone

Functional Description: dietcoin-lib is a Java library implementing the Dietcoin protocol which is an extension of the Bitcoin protocol with the goal of enhancing Bitcoin user security on resource-constrained devices such as smartphones. This piece of software is based on the bitcoinj library, which is the most popular Java implementation of Bitcoin notably used for Android applications. dietcoin-lib also enables a block-by-block replay of the existing Bitcoin blockchain to simulate a Dietcoin client downloading the blockchain from a Dietcoin server, both ends using dietcoin-lib, with the goal of evaluation the protocole on a large real dataset.

URL: https://gitlab.inria.fr/wide/dietcoin

Publications: hal-02315154, tel-01964628, hal-01743995

Contacts: Pierre-Louis Roman, Davide Frey, François Taiani, Marc Makkes, Spyros Voulgaris

Participants: Pierre-Louis Roman, Davide Frey, François Taiani, Marc Makkes, François Taiani

6.1.3 Basalt

Keywords: Peer-sampling, Blockchain

Functional Description: A number of novel blockchain and cryptocurrency implementations rely on random peer sampling. But existing protocols remain vulnerable to Sybil attacks. BASALT is a peer-sampling protocol that addresses this limitation by leveraging three main components. First it employs a novel sampling approach, termed stubborn chaotic search, that exploits ranking functions to define a dynamic target random graph (i.e. a set of v target neighbors for each node) that cannot be controlled by Byzantine nodes. Second, it adopts a hit-counter mechanism that favors the exploration of new peers even in the presence of Byzantine nodes that flood the network with their identities. Finally, it incorporates hierarchical ranking functions that ensure that nodes sample their peers from a variety of address prefixes. The first two mechanisms ensure that the number of Byzantine nodes in a node's view cannot be increased arbitrarily by attackers. This offers protection from general Byzantine behaviors including those resulting from botnet attacks, as defined above. The third mechanism ensures that nodes sample their peers from a variety of address prefixes, thereby countering institutional attacks where the attacker controls a limited number of entire address prefixes.

Contacts: Alex Auvolat-Bernstein, Davide Frey, François Taiani, David Bromberg

7 New results

7.1 Covid-19-related results

7.1.1 Coronasurveys

Participants Davide Frey.

The CoronaSurveys project is a collaborative endeavour from several universities and research institutions (team members) Data about COVID-19 cases is collected via anonymous open surveys. The project started from the observation that national governments have problems evaluating the reach of the epidemic, due to having limited resources and tests at their disposal. This problem is especially acute in low and middle-income countries (LMICs). Hence, any simple, cheap and flexible means of evaluating the incidence and evolution of the epidemic in a given country with a reasonable level of accuracy is useful. In this work, we propose a technique based on (anonymous) surveys in which participants report on the health status of their contacts. This indirect reporting technique, known in the literature as network scale-up method, preserves the privacy of the participants and their contacts, and collects information from a larger fraction of the population (as compared to individual surveys). This technique has been deployed in the CoronaSurveys project, which has been collecting reports for the COVID-19 pandemic for more than two months. Results obtained by CoronaSurveys show the power and flexibility of the approach, suggesting that it could be an inexpensive and powerful tool for LMICs. This work was carried out in the context of a collaboration led by IMDEA Spain and involving dozens of researchers from countries across the world. Preliminary results were presented at The KDD Workshop on Humanitarian Mapping [48]. The CoronaSurveys team was also finalist in the Covid-19 Facebook Data Challenge and classified for the second phase of the X-Prize Pandemic Response Challenge.

7.2 Blockchain and Large-Scale Systems

7.2.1 Money Transfer Made Simple

Participants Alex Auvolat, Davide Frey, Michel Raynal, François Taïani.

It has recently been shown ([67]) that, contrarily to a common belief, money transfer in the presence of faulty (Byzantine) processes does not require strong agreement such as consensus. In this work [17], we go one step further by showing that money transfers do not need to explicitly capture the causality

relation that links individual transfers. A simple FIFO order between each pair of processes is sufficient. To this end, this article presents a generic money transfer algorithm that can be instantiated in both the crash failure model and the Byzantine failure model. The genericity dimension lies in the underlying reliable broadcast abstraction which must be suited to the appropriate failure model. Interestingly, whatever the failure model, the money transfer algorithm only requires adding a single sequence number to its messages as control information. Moreover, as a side effect of the proposed algorithm, it follows that money transfer is a weaker problem that the construction of a read/write register in the asynchronous message-passing crash-prone model.

7.2.2 Atomic Appends in Asynchronous Byzantine Distributed Ledgers

Participants Michel Raynal.

A Distributed Ledger Object (DLO) is a concurrent object that maintains a totally ordered sequence of records, and supports two operations: APPEND, which appends a record at the end of the sequence, and GET, which returns the whole sequence of records. This work [28] comprises two main contributions. The first contribution is a formalization of aByzantine-tolerantDistributed Ledger Object(BDLO), which is a DLO in which clients and servers processes may deviate arbitrarily from their intended behavior (i.e. they may be Byzantine). The proposed formal definition is accompanied by algorithms that implementBDLOs on top of an underlying Byzantine Atomic Broadcast service. The second contribution is a suite of algorithms, based on the previous BDLO implementations, that solve the Atomic Appends problem in the presence of asynchrony, Byzantine clients and Byzantine servers. This problem occurs when clients have a composite record (set of basic records) to append to different BDLOs, in such a way that either each basic record is appended to its BDLO (and this must occur in good circumstances),or no basic record is appended. Distributed algorithms are presented, which solve the Atomic Appends problem when the clients (involved in theAtomic Appends) and the servers (which maintain the BDLOs) may be Byzantine.

This work was performed in collaboration with Vicent Cholvi from Universitat Jaume I, Castellón, Antonio Fernandez Anta from IMDEA - Instituto Madrileño de Estudios Avanzados, Chryssis Georgiou and Nicolas Nicolaou from University of Cyprus, Cyprus.

7.2.3 Modular and Distributed IDE

Participants Alex Auvolat, Yérom-David Bromberg, François Taïani.

Integrated Development Environments (IDEs) are indispensable companions to programming languages. They are increasingly turning towards Web-based infrastructure. The rise of a protocol such as the Language Server Protocol (LSP) that standardizes the separation between a language-agnostic IDE, and a language server that provides all language services (e.g., auto completion, compiler...) has allowed the emergence of high quality generic Web components to build the IDE part that runs in the browser. However, all language services require different computing capacities and response times to guarantee a user-friendly experience within the IDE. The monolithic distribution of all language services prevents to leverage on the available execution platforms (e.g., local platform, application server, cloud). In contrast with the current approaches that provide IDEs in the form of a monolithic client-server architecture, we explore in this work [29] the modularization of all language services to support their individual deployment and dynamic adaptation within an IDE. We evaluate the performance impact of the distribution of the language services across the available execution platforms on four EMF-based languages, and demonstrate the benefit of a custom distribution.

This work was done in collaboration with Fabien Coulon, Benoit Combemale, Olivier Barais, and Noël Plouzeau from the DIVERSE team.

7.2.4 DroidAutoML: A Microservice architecture to Automate the evaluation of Android Machine Learning Detection Systems

Participants Yérom-David Bromberg, Louison Gitzinger.

The mobile ecosystem is witnessing an unprecedented increase in the number of malware in the wild. To fight this threat, actors from both research and industry are constantly innovating to bring concrete solutions to improve security and malware protection. Traditional solutions such as signature-based anti viruses have shown their limits in front of massive proliferation of new malware, which are most often only variants specifically designed to bypass signature-based detection. Accordingly, it paves the way to the emergence of new approaches based on Machine Learning (ML) technics to boost the detection of unknown malware variants. Unfortunately, these solthiutions are most often under-exploited due to the time and resource costs required to adequately fine tune machine learning algorithms. In reality, in the Android community, state-of-the-art studies do not focus on model training, and most often go through an empirical study with a manual process to choose the learning strategy, and/or use default values as parameters to configure ML algorithms. However, a generic and scalable solution to automatically both configure and evaluate ML algorithms to efficiently detect Android malware detection systems. In this work [40], we introduce our approach which is based on devOps principles and a microservice architecture deployed over a set of nodes to scale and exhaustively test a large number of ML algorithms and hyper-parameters combinations. We are able to systematically find the best fit to increase up to 11% the accuracy of two state-of-the-art Android malware detect a generic and scalable solution to automatically both configure and evaluate ML algorithms to efficiently detect Android malware detection systems.

7.2.5 Foiling Sybils with HAPS in Permissionless Systems: An Address-based Peer Sampling Service

Participants Amaury Bouchra-Pilet, Davide Frey, François Taiani.

Blockchains and distributed ledgers have brought renewed interest in Byzantine fault-tolerant protocols and decentralized systems, two domains studied for several decades. Recent promising works have in particular proposed to use epidemic protocols to overcome the limitations of popular Blockchain mechanisms, such as proof-of-stake or proof-of-work. These works unfortunately assume a perfect peer-sampling service, immune to malicious attacks, a property that is difficult and costly to achieve. In this work [39], we revisited this fundamental problem with a novel Byzantine-tolerant peer-sampling service that is resilient to Sybil attacks in open systems by exploiting the underlying structure of wide-area networks.

7.2.6 DiagSys: network and third-party web-service monitoring from the browser's perspective

Participants Loïck Bonniot, François Taïani.

Internet Service Providers, on-line service providers and their end-users need accurate and automated tools to measure and diagnose networks and third-party on-line services on a large scale. To provide insightful reports, such tools should ideally reflect the Quality of Experience (QoE) perceived by end-users when they use on-line services such as websites and web APIs. Because QoE problems are often explained by causes near end users, many past measurement approaches have been implemented at the network's edge, by taking the viewpoint of either the home gateway, the browser, or by using dedicated tools running on end-user devices. In this work [26], we propose to take stock of these seminal approaches to get one step closer to a *holistic* monitoring of QoE conditions: we combine end-user perspective with

infrastructure-side insights in a more systematic monitoring strategy, which is often lacking in previous solutions.

More concretely, we argue that although the location of measuring probes in the network is critical, the device used (PC, smartphone ...) and the execution environment are also essential to capture a user's QoE. We therefore advocate that measurements should whenever possible be taken from end-user devices. This implies that any user-side measurement software should be easy to deploy and use, remain non-intrusive and incur a minimal network overhead. Browser-based measurements—the approach we explore in the paper—adhere to the above principles.

We present DIAGSYS, a crowd-sourced data collection system targeted at monitoring networks and third party web-services. DIAGSYS has been deployed online[2] and is used by volunteers to collect network metrics. It combines browser-based probes, running both on end-user devices and in headless browsers, and landmark servers hosting measurement services. Our browser-based probes are compatible with the recent security restrictions of modern browsers, and systematically monitor a set of pre-configured services. We describe a first set of case studies based on the data collected so far.

This work has been done in collaboration with Christoph Neumann (InterDigital).

7.3 Scaling and understanding AI systems

7.3.1 FeGAN: Scaling Distributed GANs

Participants Erwan Le Merrer.

Existing approaches to distribute Generative Adversarial Networks (GANs) either(i) fail to scale for they typically put the two components of a GAN (the generator and the discriminator) on different machines, inducing significant communication overhead, or (ii) they face GAN training specific issues, exacerbated by distribution. In this work [34], we propose FeGAN, the first middleware for distributing GANs over hundreds of devices addressing the issues of mode collapse and vanishing gradients. Essentially, we revisit the idea of Federated Learning, co-locating a generator with a discriminator on each device (addressing the scaling problem) and having a server aggregate the devices' models using balanced sampling and Kullback-Leibler (KL) weighting, mitigating training issues and boosting convergence. Through extensive experiments, we show that FeGAN generates high–quality dataset samples in a scalable and devices' heterogeneity tolerant manner. In particular,FeGAN achieves up to 5× throughput gain with 1.5× less bandwidth compared to the state–of–the–art GAN distributed approach (named MD–GAN), while scaling to at least one order of magnitude more devices. We demonstrate that FeGAN boosts training by 2.6× w.r.t. a baseline application of Federated Learning to GANs, while preventing training issues.

This work was performed in collaboration with Rachid Guerraoui, Arsany Guirguis, Anne-Marie Kermarrec from EPFL (Lausane, Switzerland).

7.3.2 The Imitation Game: Algorithm Selection by Exploiting Black-Box Recommenders

Participants Erwan Le Merrer.

Cross-validation is commonly used to select the recommendation algorithms that will generalize best on yet unknown data. Yet, in many situations the available dataset used for cross-validation is scarce and the selected algorithm might not be the best suited for the unknown data. In contrast, established companies have a large amount of data available to select and tune their recommender algorithms, which therefore should generalize better. These companies often make their recommender systems available as black-boxes, i.e., users query the recommender through an API or a browser. This work [31] proposes RECRANK, a technique that exploits a black-box recommender system, in addition to classic cross-validation. RECRANK employs graph similarity measures to compute a distance between the

[2] https://diagnet.gitlabpages.inria.fr/

output recommendations of the black-box and of the considered algorithms. We empirically show that RECRANK provides a substantial improvement (33%) for the selection of algorithms for the MovieLens dataset, in comparison with standalone cross-validation.

This work was performed in collaboration with Georges Damaskinos and Rachid Guerraoui from EPFL (Lausanne, Switzerland) and Christoph Neuman (InterDigital, Rennes).

7.3.3 Remote Explainability faces the bouncer problem

Participants Erwan Le Merrer.

The concept of explainability is envisioned to satisfy society's demands for transparency about machine learning decisions. The concept is simple: like humans, algorithms should explain the rationale behind their decisions so that their fairness can be assessed. Although this approach is promising in a local context (for example, the model creator explains it during debugging at the time of training), we argue in this work [19] that this reasoning cannot simply be transposed to a remote context, where a model trained by a service provider is only accessible to a user through a network and its application programming interface. This is problematic, as it constitutes precisely the target use case requiring transparency from a societal perspective. Through an analogy with a club bouncer (who may provide untruthful explanations upon customer rejection), we show that providing explanations cannot prevent a remote service from lying about the true reasons leading to its decisions. More precisely, we observe the impossibility of remote explainability for single explanations by constructing an attack on explanations that hides discriminatory features from the querying user. We provide an example implementation of this attack. We then show that the probability that an observer spots the attack, using several explanations for attempting to find incoherences, is low in practical settings. This undermines the very concept of remote explainability in general.

This work was performed in collaboration with Gilles Trédan from LAAS (Toulouse, France).

7.3.4 FLeet: Online Federated Learning via Staleness Awareness and Performance Prediction

Participants François Taïani.

Federated Learning (FL) is very appealing for its privacy benefits: essentially, a global model is trained with updates computed on mobile devices while keeping the data of users local. Standard FL infrastructures are however designed to have no energy or performance impact on mobile devices, and are therefore not suitable for applications that require frequent (online) model updates, such as news recommenders. This work [30] presents FLeet, the first Online FL system, acting as a middleware between the Android OS and the machine learning application. FLeet combines the privacy of Standard FL with the precision of online learning thanks to two core components: (i) I-Prof, a new lightweight profiler that predicts and controls the impact of learning tasks on mobile devices, and (ii) AdaSGD, a new adaptive learning algorithm that is resilient to delayed updates. Our extensive evaluation shows that Online FL, as implemented by FLeet, can deliver a 2.3× quality boost compared to Standard FL, while only consuming 0.036% of the battery per day. I-Prof can accurately control the impact of learning tasks by improving the prediction accuracy up to 3 computation time) and up to 19× (energy). AdaSGD outperforms alternative FL approaches by 18.4% in terms of convergence speed on heterogeneous data.

This work was done in collaboration with Georgios Damaskinos, Rachid Guerraoui, and Anne-Marie Kermarrec from EPFL (Lausane, Switzerland), Vlad Nitu (from LIRIS/CNRS, Lyon), and Rhicheek Patra from EPFL (Lausane, Switzerland).

7.3.5 Smaller, Faster & Lighter KNN Graph Constructions

Participants François Taïani.

We propose GoldFinger [35], a new compact and fast-to-compute binary representation of datasets to approximate Jaccard's index. We illustrate the effectiveness of GoldFinger on the emblematic big data problem of K-Nearest-Neighbor (KNN) graph construction and show that GoldFinger can drastically accelerate a large range of existing KNN algorithms with little to no overhead. As a side effect, we also show that the compact representation of the data protects users' privacy for free by providing k-anonymity and l-diversity. Our extensive evaluation of the resulting approach on several realistic datasets shows that our approach delivers speedups of up to 78.9% compared to the use of raw data while only incurring a negligible to moderate loss in terms of KNN quality. To convey the practical value of such a scheme, we apply it to item recommendation and show that the loss in recommendation quality is negligible.

This work was done in collaboration with Rachid Guerraoui and Anne-Marie Kermarrec from EPFL (Lausane, Switzerland), and Olivier Ruas from Peking University (China).

7.4 Distributed Network and Graph Algorithms

7.4.1 From Bezout's Identity to Space-Optimal Election in Anonymous Memory Systems

Participants Michel Raynal.

An anonymous shared memory REG 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 REG[x] by a process p and as REG[y] (where $y \neq x$) by another process q. Moreover, the register known as REG[a] by a process p and the register known as REG[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. This work [33] focuses on solving the d-election problem, in which it is required to elect at least one and at most d leaders, in such an anonymous shared memory system. We notice that the 1-election problem is the familiar leader election problem. Let n be the number of processes and m the size of the anonymous memory (number of atomic registers). We show that the condition gcd(m, n) $\leq$ d is necessary and sufficient for solving the d-election problem, where communication is through read/write or read+modify+write registers. The algorithm used to prove the sufficient condition relies on Bezout's Identity - a Diophantine equation relating numbers according to their Greatest Common Divisor. Furthermore, in the process of proving the sufficient condition, it is shown that 1-leader election can be solved using only a single read/write register (which refutes a 1989 conjecture stating that three non-anonymous registers are necessary), and that the exact d-election problem, where exactly d leaders must be elected, can be solved if and only if gcd(m, n) divides d.

This work was performed in collaboration with Emmanuel Godard, and Damien Imbs from Aix-Marseille University, France, and G. Taubenfeld from Herzliya, Israel.

7.4.2 Mutual exclusion in fully anonymous shared memory systems

Participants Michel Raynal.

Process anonymity has been studied for a long time. Memory anonymity is more recent. In an anonymous memory system, there is no a priori agreement among the processes on the names of the shared registers. As an example, a shared register named A by a process p and a shared register named B by another process q may correspond to the very same register X, while the same name C may correspond to different register names for the processes p and q, and this remains unknown to the processes. This work [21] introduces the full anonymous model, namely a model in which both the processes and the

registers are anonymous. A fundamental question is then “is this model meaningful?”, which can be translated as “can non-trivial fundamental problems be solved in such a very weak computing model?”

In this work, we answer this question positively. More precisely, we present a deadlock-free mutual exclusion algorithm in such a fully anonymous model where the anonymous registers are read/modify/write registers. This algorithm assumes that m (the number of shared registers) and n (the number of processes) are such that m is relatively prime with all the integers . Combined with a previous result (PODC 2019) on mutual exclusion in memory anonymous (but not process anonymous) systems, it follows that this condition is both necessary and sufficient for the existence of such an algorithm in fully anonymous systems. As far as we know, this is the first time full anonymity is considered, and where a non-trivial concurrency-related problem is solved in such a very strong anonymity context.

Once election is solved, a general (and simple) de-anonymization algorithm is presented, which takes as a subroutine any memory anonymous leader election algorithm. Hence, any instance of this algorithm works for the values of m required by the selected underlying election algorithm. As the underlying election algorithms, the de-anonymization algorithm is symmetric in the sense that process identities can only be compared for equality.

This work was performed in collaboration with Gadi Taubenfeld from Interdisciplinay Center Herzliya, Israel.

7.4.3 k-Immediate Snapshot and x-Set Agreement: How Are They Related?

Participants Michel Raynal.

An immediate snapshot object is a high level communication object, built on top of a read/write distributed system in which all except one processes may crash. This object provides the processes with a single operation, denoted $write\_snapshot()$, which allows the invoking process to write a value and obtain a set of pairs <process id, value> satisfying some set containment properties, that represent a snapshot of the values written to the object, occurring immediately after the write step.

Considering an n-process model in which up to t processes may crash, this paper introduces first the k-resilient immediate snapshot object, which is a natural generalization of the basic immediate snapshot (which corresponds to the case $k = t = n-1$). In addition to the set containment properties of the basic immediate snapshot, a k-resilient immediate snapshot object requires that each set returned to a process contains at least $n-k$ pairs.

This work [41], done in collaboration with Carole Delporte, Hugues Fauconnier, and Sergio Rajsbaum, first shows that, for $k, t < n-1$, k-resilient immediate snapshot is impossible in asynchronous read/write systems. Then it investigates a model of computation where the processes communicate with each other by accessing k-immediate snapshot objects, and shows that this model is stronger than the t-crash model. Considering the space of x-set agreement problems (which are impossible to solve in systems such that $x \leq t$), our results show then that x-set agreement can be solved in read/write systems enriched with k-immediate snapshot objects for $x = max(1, t+k-(n-2))$. They also show that, in these systems, k-resilient immediate snapshot and consensus are equivalent when $1 \leq t < n/2$ and $t \leq k \leq (n-1)-t$. Hence, we establish strong relations linking fundamental distributed computing objects (one related to communication, the other to agreement), which are impossible to solve in pure read/write systems.

7.4.4 An Eventually Perfect Failure Detector for Networks of Arbitrary Topology Connected with ADD Channels Using Time-To-Live Values

Participants Michel Raynal.

We introduced an eventually perfect failure detector in an arbitrarily connected, partitionable network. We assume ADD channels: for each one there exist constants K, D, not known to the processes, such that for every K consecutive messages sent in one direction, at least one is delivered within time D. The best

previous implementation used messages of bounded size, but exponential in n, the number of nodes. The main contribution of this work [22], done in collaboration with Karla Vargas and Sergio Rasjbaum, is a novel use of time-to-live values in the design of failure detectors, obtaining a flexible implementation that uses messages of size O(n log n).

7.4.5 Self-stabilizing Uniform Reliable Broadcast

Participants Michel Raynal.

This work [38] studies a well-known communication abstraction called Uniform Reliable Broadcast (URB). URB is central in the design and implementation of fault-tolerant distributed systems, as many non-trivial fault-tolerant distributed applications require communication with provable guarantees on message deliveries. Our study focuses on fault-tolerant implementations for time-free message-passing systems that are prone to node-failures. Moreover, we aim at the design of an even more robust communication abstraction. We do so through the lenses of self-stabilization—a very strong notion of fault-tolerance. In addition to node and communication failures, self-stabilizing algorithms can recover after the occurrence of arbitrary transient faults; these faults represent any violation of the assumptions according to which the system was designed to operate (as long as the algorithm code stays intact. We propose the first self-stabilizing URB algorithm for asynchronous (time-free) message-passing systems that are prone to node-failures. This work was done in collaboration with Oskar Lundström and Elad Schiller.

7.4.6 60 Years of Mastering Concurrent Computing through Sequential Thinking

Participants Michel Raynal.

Modern computing systems are highly concurrent. Threads run concurrently in shared-memory multi-core systems, and programs run in different servers communicating by sending messages to each other. Concurrent programming is hard because it requires to cope with many possible, unpredictable behaviors of the processes, and the communication media. Right from the start in 1960's, the main way of dealing with concurrency has been by reduction to sequential reasoning. In this work [20], done in collaboration with Sergio Rasjbaum, we traced this history, and illustrated it through several examples, from early ideas based on mutual exclusion (which was initially introduced to access shared physical resources), passing through consensus and concurrent objects (which are immaterial data), until today distributed ledgers. We also discussed on the limits that this approach encounters, related to fault-tolerance, performance, and inherently concurrent problems.

7.4.7 Collisions Are Preferred: RFID-Based Stocktaking with a High Missing Rate

Participants Michel Raynal.

RFID-based stocktaking uses RFID technology to verify the presence of objects in a region e.g., a warehouse or a library, compared with an inventory list. The existing approaches for this purpose assume that the number of missing tags is small. This is not true in some cases. For example, for a handheld RFID reader, only the objects in a larger region (e.g., the warehouse) rather than in its interrogation region can be known as the inventory list, and hence many tags in the list are regarded as missing. The missing objects significantly increase the time required for stocktaking. In this work [23], done in collaboration with Weiping Zhu, Xing Meng, Xiaolei Peng and Jiannong Cao, we propose an algorithm called CLS (Coarse-grained inventory list based stocktaking) to solve this problem. CLS enables multiple missing objects to hash to a single time slot and thus verifies them together. CLS also improves the existing approaches by

utilizing more kinds of RFID collisions and reducing approximately one-fourth of the amount of data sent by the reader. Moreover, we observe that the missing rate constantly changes during the identification because some of tags are verified present or absent, which affects time efficiency; accordingly, we propose a hybrid stocktaking algorithm called DLS (Dynamic inventory list based stocktaking) to adapt to such changes for the first time. According to the results of extensive simulations, when the inventory list is 20 times that of actually present tags, the execution time of our approach is 36.3 percent that of the best existing algorithm.

7.4.8 Optimal time and space leader election in population protocols

Participants George Giakkoupis.

Population protocols are a model of distributed computing, where n agents with limited computational power and memory perform randomly scheduled pairwise interactions. A fundamental problem in this setting is that of leader election, where all agents start from the same state, and they seek to reach and maintain a global state where exactly one agent is in a dedicated leader state.

A significant amount of work has been devoted to the study of the time and space complexity of this problem. Alistarh et al. (SODA'17) have shown that $\Omega(\log\log n)$ states per agent are needed in order to elect a leader in fewer than $\tilde{\Theta}(n^2)$ expected interactions. Moreover, $\Omega(n\log n)$ expected interactions are required regardless of the number of states (Sudo and Masuzawa; 2020). On the upper bound side, Gasieniec and Stachowiak (SODA'18) have presented the first protocol that uses an optimal, $\Theta(\log\log n)$, number or states and elects a leader in $O(n\log^2 n)$ expected interactions. This running time was subsequently improved to $O(n\log n\log\log n)$ (Gasieniec et al.; SPAA'19).

In [25] we provide the first leader election population protocol that is both time and space optimal: it uses $\Theta(\log\log n)$ states per agent, and elects a leader in $O(n\log n)$ interactions in expectation. A key novel component of our approach is a simple protocol that efficiently selects a small set of agents of poly$(\log n)$ size, given an initial set of $s = O(n^\epsilon)$ selected agents. Unlike existing approaches, which proceed by shrinking the initial set monotonically over time, our novel component first increases the set in a controlled way to a specific size (which is independent of s) before it shrinks the set to a poly$(\log n)$ size.

This is a joitn work with Petra Berenbrink and Peter Kling from U. Hamburg.

7.4.9 Spread of information and diseases via random walks in sparse graphs

Participants George Giakkoupis.

In [32] we consider a natural network diffusion process, modeling the spread of information or infectious diseases. Multiple mobile agents perform independent simple random walks on an n-vertex connected graph G. The number of agents is linear in n and the walks start from the stationary distribution. Initially, a single vertex has a piece of information (or a virus). An agent becomes informed (or infected) the first time it visits some vertex with the information (or virus); thereafter, the agent informs (infects) all vertices it visits. Giakkoupis et al. (PODC'19) have shown that the spreading time, i.e., the time before all vertices are informed, is asymptotically and w.h.p. the same as in the well-studied randomized rumor spreading process, on any d-regular graph with $d = \Omega(\log n)$. The case of sub-logarithmic degree was left open, and is the main focus of this paper. First, we observe that the equivalence shown by Giakkoupis et al. does not hold for small d: We give an example of a 3-regular graph with logarithmic diameter for which the expected spreading time is $\Omega(\log^2 n/\log\log n)$, whereas randomized rumor spreading is completed in time $\Theta(\log n)$, w.h.p. Next, we show a general upper bound of $\tilde{O}(d\cdot diam(G)+\log^3 n/d)$, w.h.p., for the spreading time on any d-regular graph. We also provide a version of the bound based on the average degree, for non-regular graphs. Next, we give tight analyses for specific graph families. We show that the spreading time is $O(\log n)$, w.h.p., for constant-degree regular expanders. For the binary tree, we show an

upper bound of $O(\log n \cdot \log\log n)$, w.h.p., and prove that this is tight, by giving a matching lower bound for the cover time of the tree by n random walks. Finally, we show a bound of $O(diam(G))$, w.h.p., for k-dimensional grids, by adapting a technique by Kesten and Sidoravicius.

This is a joitn work with Hayk Saribekyan and Thomas Sauerwald from U. Cambridge.

7.4.10 Self-stabilizing clock synchronization with 1-bit messages

Participants George Giakkoupis.

In [24] we study the fundamental problem of distributed clock synchronization in a basic probabilistic communication setting. We consider a synchronous fully-connected network of n agents, where each agent has a local clock, that is, a counter increasing by one modulo T in each round. The clocks have arbitrary values initially, and they must all indicate the same time eventually. We assume a pull communication model, where in every round each agent receives an ℓ-bit message from a random agent. We devise several fast synchronization algorithms that use small messages and are self-stabilizing, that is, the complete initial state of each agent (not just its clock value) can be arbitrary.

We first provide a surprising algorithm for synchronizing a binary clock ($T = 2$) using 1-bit messages ($\ell = 1$). This is a variant of the voter model and converges in $O(\log n)$ rounds w.h.p., unlike the voter model which needs polynomial time. Next we present an elegant extension of our algorithm that synchronizes a modulo $T = 4$ clock, with $\ell = 1$, in $O(\log n)$ rounds. Using these two algorithms, we refine an algorithm of Boczkowski et al. (SODA'17), that synchronizes a modulo T clock in polylogarithmic time (in n and T). The original algorithm uses $\ell = 3$ bit messages, and each agent receives messages from two agents per round. Our algorithm reduces the message size to $\ell = 2$, and the number of messages received to one per round, without increasing the running time. Finally, we present two algorithms that simulate our last algorithm achieving $\ell < 2$, without hurting the asymptotic running time. The first algorithm uses a message space of size 3, i.e., $\ell = \log_2(3)$. The second requires a rough upper bound on $\log n$, and uses just 1-bit messages. More generally, our constructions can simulate any self-stabilizing algorithm that requires a shared clock, without increasing the message size and by only increasing the running time by a constant factor and a polylogarithmic term.

This is a joitn work with Paul Bastide from ENS Rennes and Hayk Saribekyan from U. Cambridge.

8 Bilateral contracts and grants with industry

8.1 Bilateral contracts with industry

CIFRE InterDigital: Distributed troubleshooting of edge-compute functions (2018-2021)

Participants Loïck Bonniot, François Taïani.

This project seeks to explore how recent generations of end-user gateways (or more generally end-user devices) could implement an edge-compute paradigm powered by user-side micro-services. Our vision is that the devices distributed among the homes of end-users will expose (as a service) their computing power and their ability to quickly deploy compute functions in an execution environment. In order for service and application providers to actually use the system and deploy applications, the system must however ensure an appropriate level of reliability, while simultaneously requiring a very low level of maintenance in order to address the typical size and economics of gateway deployments (at least a few tens of million units). Providing a good level of reliability in such a large system at a reasonable cost is unfortunately difficult. To address this challenge, we aim in this thesis to exploit the *natural distribution* of such large-scale user-side device deployments to quickly pinpoint problems and troubleshoot applications experiencing performance degradations.

This project is in collaboration with Christoph Neumann from InterDigital.

9 Partnerships and cooperations

9.1 International initiatives

9.1.1 Inria international partners

Informal international partners

- The LPD lab (Distributed Programming Laboratory) from EPFL (Lausane, Switzerland, Prof. Rachid Guerraoui)
- The SACS lab (Systèmes distribués large échelle) from EPFL (Lausane, Switzerland, Prof. Anne-Marie Kermarrec)

9.2 International research visitors

9.2.1 Visits to international teams

Research stays abroad

- Erwan Le Merrer was on a research visit at EPFL, with Prof. Anne-Marie Kermarrec, in September 2020. Collaboration on graph watermarking.

9.3 European initiatives

9.3.1 Collaborations with major European organizations

- Collaboration of Davide Frey with IMDEA Madrid and several other European universities in the Coronasurveys project.

9.4 National initiatives

ANR Project PAMELA (2016-2022)

Participants Davide Frey, George Giakkoupis, François Taïani.

PAMELA is a collaborative ANR project involving Inria/IRISA, Inria Lille (MAGNET team), UMPC, Mediego and Snips. The project aims at developing machine learning theories and algorithms in order to learn local and personalized models from data distributed over networked infrastructures. This project seeks to provide fundamental answers to modern information systems built by interconnecting many personal devices holding private user data in the search of personalized suggestions and recommendations. A significant asset of the project is the quality of its industrial partners, Snips and Mediego, who bring in their expertise in privacy protection and distributed computing as well as use cases and datasets.

ANR Project OBrowser (2016-2021)

Participants David Bromberg, Davide Frey, François Taïani.

OBrowser is a collaborative ANR project involving Inria, the University of Nantes, the University of South Brittany, and Orange. The project emerges from the vision of designing and deploying distributed applications on millions of machines using web-enabled technologies without relying on a cloud or a central authority. OBrowser proposes to build collaborative applications through a decentralized execution environment composed of users' browsers that autonomously manages issues such as communication, naming, heterogeneity, and scalability.

ANR Project DESCARTES (2016-2021)

Participants George Giakkoupis, Michel Raynal, François Taïani.

DESCARTES is a collaborative ANR project involving Inria/IRISA, Labri (U. Bordeaux), IRIF (U. Paris Diderot), Inria Paris (GANG Team), Vérimag (Grenoble), LIF (Marseilles), and LS2N (former LINA, Nantes). The DESCARTES project aims at bridging the lack of a generic theoretical framework in order to unify the large body of fundamental knowledge on distributed computation that has been acquired over the last 40 years. In particular, the project's objective is to develop a systematic model of distributed computation that organizes the functionalities of a distributed computing system into reusable modular constructs assembled via well-defined mechanisms that maintain sound theoretical guarantees on the resulting system.

Task force for the DGE

Participants Erwan Le MErrer.

Erwan Le Merrer acted as an expert for a task force for the ministry of economy (DGE) in June 2020, on algorithm transparency and algorithmic regulation. Co-produced a draft for preparing the European Digital Services Act.

10 Dissemination

10.0.1 Scientific events: organisation

Member of the organizing committees

- François Taïani served as Tutorial Co-Chair of the 50th IEEE/IFIP Int. Conference on Dependable Systems and Networks, June 29-July 2 2020, Valencia, Spain (DSN 2020)
- François Taïani served ad Publicity Co-Chair of 39th International Symposium on Reliable Distributed Systems, Sep 21-24, 2020, Shanghai, China (SRDS 2020)
- François Taïani was Co-Organizer of the 4th Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers, Dec 8, 2020, Delft, The Netherlands, Colocated with Middleware 2020 (SERIAL 2020)

10.0.2 Scientific events: selection

Member of the conference program committees

- François Taïani served on the PC of the 40th IEEE International Conference on Distributed Computing Systems. Nov. 29 - Dec. 1, 2020, Singapore (ICDCS 2020)
- François Taïani served on the PC of the ACM/IFIP International Conference on Middleware 2020, Dec. 7-11 2020, Delft, The Netherlands (Middleware 2020)
- François Taïani served on the PC of the 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing, May 11, 2020 - May 14, 2020, Melbourne, Australia (CCGrid 2020)
- François Taïani served on the PC of the 2nd International Conference on Blockchain Economics, Security and Protocols, Oct. 26-27 2020, Toulouse, France (Tokenomics 2020)
- Davide Frey served on the PC of the 20th International Conference on Distributed Applications and Interoperable Systems . June 15 - 19, 2020, (DAIS 2020)

- Davide Frey served on the PC of the 4th ACM International Conference on Distributed and Event-Based Systems, July 13 – 17 2020 (DEBS 2020)
- Davide Frey served in the PC of the 4th Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers, Dec 8, 2020, Delft, The Netherlands, Colocated with Middleware 2020 (SERIAL 2020)
- Davide Frey served in the PC of the African Conference on Research in Computer Science and Applied Mathematics (CARI 2020)

Reviewer

- Davide Frey was a external reviewer for ICDCS 2020.

10.0.3 Journal

Member of the editorial boards

Reviewer - reviewing activities

- George Giakkoupis was a reviewer for Algorithmica (ALGO).
- George Giakkoupis was a reviewer for Transactions on Parallel and Distributed Systems (TPDS)
- George Giakkoupis was a reviewer for Distributed Computing (DIST)
- George Giakkoupis was a reviewer for Transactions on Knowledge and Data Engineering (TKDE)
- François Taïani served as reviewer of the 17th USENIX Symposium on Networked Systems Design and Implementation, February 25–27, 2020, Santa Clara, CA, USA (NSDI 2020)
- François Taïani served as reviewer for IEEE Transactions on Parallel and Distributed Systems (TPDS).
- François Taïani served as reviewer for IEEE Transactions on Knowledge and Data Engineering (TKDE).
- Davide Frey served as reviewer for the Journal of Parallel and Distributed Computing (JPDC).
- Davide Frey served as reviewer for the Internet Computing Journal (IC).
- Davide Frey served as reviewer for Transactions on Network Science and Engineering (TNSE)
- Davide Frey served as reviewer for the STIC-AmSud Program

10.0.4 Invited talks

- George Giakkoupis. An optimal leader election population protocol. Algorithms Seminar, University of Wrocław, Poland, Jun. 17 2020
- George Giakkoupis. An optimal leader election population protocol. 2nd Workshop on Distributed Algorithms for Low-Functional Robots (WDALFR), Nara, Japan, Jan. 8 2021
- Erwan Le Merrer was invited to give a talk at the THCON 2020 on March 10 in Toulouse, entitled “The topological face of recommendation”.
- David Bromberg. A generative approach to circumvent existing malware detection systems. GDR RSD/ASF Winter School on Distributed systems and networks , Pleynet, France, 6 Février 2020

10.0.5 Leadership within the scientific community

- François Taïani has been a member of the Gilles Kahn PhD Award in Computer Science, from Société Informatique de France (SIF) since 2018.
- David Bromberg is responsible of the large scale, distributed system department at the research lab IRISA, IRISA, Rennes Bretagne Atlantique since 2020.

10.0.6 Research administration

- François Taïani is a Career Advice Person, (Référent conseil-parcours professionel chercheurs) for IRISA/Inria Rennes Bretagne Atlantique since 2019.
- David Bromberg is responsible of international relationships of IRISA, IRISA, Rennes Bretagne Atlantique from 2016 to 2020.

10.1 Teaching - Supervision - Juries

Teaching

- Master: Florestan De Moor, Programmation Dirigée par la Syntaxe, 22h, M1-SIF, Université de Rennes 1, France
- Engineering School: François Taïani, Synchronization and Parallel Programming, 34h, 2nd year of Engineering School (M1), ESIR / Univ. Rennes I, France.
- Engineering School: François Taïani, Distributed Systems, 24h, 3rd year of Engineering School (M2), ESIR / Univ. Rennes I, France.
- Engineering School: François Taïani, Introduction to Operating Systems, 24h, 1st year of Engineering School (L3), ESIR / Univ. Rennes I, France.
- Bacherlor: François Taïani, Distributed Algorithms, 20h, L3 Parcours SI, ENS Rennes, France.
- Master: Davide Frey, Scalable Distributed Systems, 10 hours, M1, EIT/ICT Labs Master School, Univ. Rennes I, France.
- Master: Davide Frey, Big-Data Storage and Processing Infrastructures, 10 hours, M2-SIF, Univ. Rennes I, France.
- Master: Davide Frey, Cloud Computing, 6 hours, M2-MIAGE, Univ. Rennes I, France.
- Master: Davide Frey, Distributed Systems/Systèmes Répartis, 12 hours, ENSAI, France.
- Master: Davide Frey, Apprentice Tutoring, 8 ETD hours, M2 Alternance Univ. Rennes I, France.
- Master: Quentin Dufour and Davide Frey, 12 ETD hours, projet M1 ISTIC, Univ. Rennes I, France.
- Master / PhD: Davide Frey, Distributed Computing and Blockchain, 30 hours, UM6P, Morocco.
- Master: Erwan Le Merrer, Projet , 36 ETD hours, M1 ISTIC, Univ. Rennes I, France.
- Master: Erwan Le Merrer, Network Science, 12 hours, M2 ESIR, Univ. Rennes I, France.
- Master: Quentin Dufour and Louison Gitzinger, Cloud, 36 hours, M2 ESIR, Univ. Rennes I, France.
- Master: Quentin Dufour, TLC, 18 hours, M2 ISTIC, Univ. Rennes I, France
- Master: George Giakkoupis, Distributed Systems/Systèmes Répartis, 9 hours, ENSAI Rennes, France.
- Engineering School: David Bromberg, Web for Internet of things, 40h, 2nd year of Engineering School (M1), ESIR / Univ. Rennes I, France.

- Engineering School: David Bromberg, Network and system security for Internet of things, 40h, 2nd year of Engineering School (M1), ESIR / Univ. Rennes I, France.
- Engineering School: David Bromberg, IoT Project , 50h, 2nd year of Engineering School (M1), ESIR / Univ. Rennes I, France.
- Engineering School: David Bromberg, Cloud , 10h, 3rd year of Engineering School (M2), ESIR / Univ. Rennes I, France.
- Master: David Bromberg, System programming, 20h, 2nd year of Engineering School (M1), Polytechnique, Cameroun.
- Engineering School: David Bromberg, Head of the IoT diploma, Engineering School, ESIR / Univ. Rennes I, France.

Supervision

- PhD in progress: Quentin Dufour, BBDA - Browser Based Data Analytics, January 2018, David Bromberg and Davide Frey.
- PhD in progress: Amaury Bouchra Pilet, Robust and Lightweight Overlay Management for Decentralized Learning, University of Rennes 1, September 2018, David Bromberg and Davide Frey.
- PhD in progress: Loïck Bonniot, Distributed Troubleshooting of Edge-Compute Functions, University of Rennes 1, François Taïani and Christoph Neumann (Interdigital, CIFRE).
- PhD in progress: Alex Auvolat, Towards probabilistic decentralized systematic design for large-scale privacy-preserving collaborative systems, University of Rennes 1, François Taïani and David Bromberg.
- PhD in progress: Hayk Saribekyan, Randomized Algorithms for Distributed Information Dissemination, University of Cambridge, UK, George Giakkoupis and Thomas Sauerwald (U. Cambridge).
- PhD in progress: Thibault Maho, Black-box attacks on neural models, Erwan Le Merrer.
- PhD :Louison Gitzinger, Surviving the massive proliferation of mobile malware, University of Rennes, 08 Décembre 2020, David Bromberg

Juries

- François Taïani was an examiner for Zoltan Miklos' HDR thesis: From databases to artificial intelligence, Université de Rennes 1 (France), 6 March 2020
- François Taïani was an examiner for Emmanuelle Anceaume's ' HDR thesis: Abstractions to Build Permissionless Distributed Systems, Université de Rennes 1 (France), 18 December 2020
- François Taïani was a reviewer for George Damaskinos's PhD thesis: Private And Secure Distributed Learning, EPFL (Switzerland), 19 May 2020
- Erwan Le Merrer was a jury member of the EDIT doctoral school in Paris for Ph.D. grants in 2020.
- David Bromberg was an external reviewer for Mathias Lacaud's PhD thesis: Towards pragmatic solutions to improve the quality of video streamming in current and future networks, University of Bordeaux, 8 October 2020.

10.2 Outreach

10.2.1 Internal or external Inria responsibilities

- Davide Frey was scientific correspondant for the DPEI until Dec 2020.

11 Scientific production

11.1 Major publications

[1] P. Berenbrink, G. Giakkoupis and P. Kling. 'Tight Bounds for Coalescing-Branching Random Walks on Regular Graphs'. In: *SODA 2018 - Proceedings of the 29th ACM-SIAM Symposium on Discrete Algorithms*. New Orleans, United States: ACM, Jan. 2018, pp. 1715–1733. URL: https://hal.inria.fr/hal-01635757.

[2] L. Bonniot, C. Neumann and F. Taïani. 'DiagSys: network and third-party web-service monitoring from the browser's perspective (industry track)'. In: *2020 - ACM/IFIP Middleware*. Delft, Netherlands: ACM, Dec. 2020, pp. 1–7. URL: https://hal.archives-ouvertes.fr/hal-02967290.

[3] S. Bouget, Y.-D. Bromberg, A. Luxey and F. Taïani. 'Pleiades: Distributed Structural Invariants at Scale'. In: *DSN 2018 - IEEE/IFIP International Conference on Dependable Systems and Networks*. Luxembourg, Luxembourg: IEEE, June 2018, pp. 542–553. DOI: 10.1109/DSN.2018.00062. URL: https://hal.archives-ouvertes.fr/hal-01803881.

[4] Z. Bouzid, M. Raynal and P. Sutra. 'Anonymous obstruction-free (n, k)-set agreement with n-k+1 atomic read/write registers'. In: *Distributed Computing* 31.2 (Apr. 2018), pp. 99–117. URL: https://hal.inria.fr/hal-01952626.

[5] Y.-D. Bromberg, Q. Dufour and D. Frey. 'Multisource Rumor Spreading with Network Coding'. In: *INFOCOM 2019 - IEEE International Conference on Computer Communications*. Paris, France: IEEE, Apr. 2019, pp. 1–10. URL: https://hal.inria.fr/hal-01946632.

[6] Y.-D. Bromberg, A. Luxey and F. Taïani. 'CASCADE: Reliable Distributed Session Handoff for Continuous Interaction across Devices'. In: *ICDCS 2018 - 38th IEEE International Conference on Distributed Computing Systems*. Vienna, Austria: IEEE, July 2018, pp. 244–254. DOI: 10.1109/ICDCS.2018.00033. URL: https://hal.inria.fr/hal-01797548.

[7] K. Censor-Hillel, M. Ghaffari, G. Giakkoupis, B. Haeupler and F. Kuhn. 'Tight Bounds on Vertex Connectivity Under Sampling'. In: *ACM Transactions on Algorithms* 13.2 (May 2017), 19:1–19:26. DOI: 10.1145/3086465. URL: https://hal.inria.fr/hal-01635743.

[8] F. Chierichetti, G. Giakkoupis, S. Lattanzi and A. Panconesi. 'Rumor Spreading and Conductance'. In: *Journal of the ACM (JACM)* 65.4 (Aug. 2018), 17:1–17:21. DOI: 10.1145/3173043. URL: https://hal.inria.fr/hal-01942162.

[9] V. Cholvi, A. Fernandez Anta, C. Georgiou, N. Nicolaou and M. Raynal. 'Atomic Appends in Asynchronous Byzantine Distributed Ledgers'. In: *2020 16th European Dependable Computing Conference (EDCC)*. Munich, Germany: IEEE, Sept. 2020, pp. 77–84. DOI: 10.1109/EDCC51268.2020.00022. URL: https://hal.inria.fr/hal-03148616.

[10] G. Damaskinos, R. Guerraoui, E. Le Merrer and C. Neumann. 'The Imitation Game: Algorithm Selection by Exploiting Black-Box Recommenders'. In: *NETYS 2020 - 8th International Conference on Networked Systems*. Marrakech / Virtual, Morocco, June 2020. URL: https://hal.archives-ouvertes.fr/hal-03118263.

[11] R. Guerraoui, A. Guirguis, A.-M. Kermarrec and E. Le Merrer. 'FeGAN: Scaling Distributed GANs'. In: *ACM/IFIP Middleware 2020 - Annual ACM/IFIP Middleware conference*. Delft / Virtual, Netherlands, Dec. 2020. URL: https://hal.archives-ouvertes.fr/hal-03118260.

[12] M. Herlihy, S. Rajsbaum, M. Raynal and J. Stainer. 'From wait-free to arbitrary concurrent solo executions in colorless distributed computing'. In: *Theoretical Computer Science* 683 (June 2017), pp. 1–21. DOI: 10.1016/j.tcs.2017.04.007. URL: https://hal.inria.fr/hal-01660566.

[13] H. Lakhlef, M. Raynal and F. Taïani. 'Vertex Coloring with Communication Constraints in Synchronous Broadcast Networks'. In: *IEEE Transactions on Parallel and Distributed Systems* 30.7 (July 2019), pp. 1672–1686. DOI: 10.1109/TPDS.2018.2889688. URL: https://hal.inria.fr/hal-02376726.

[14] A. Luxey, Y.-D. Bromberg, F. M. Costa, V. Lima, R. Da Rocha and F. Taïani. 'Sprinkler: A probabilistic dissemination protocol to provide fluid user interaction in multi-device ecosystems'. In: *PerCom 2018 - IEEE International Conference on Pervasive Computing and Communications*. Athens, Greece: IEEE, Mar. 2018, pp. 1–10. DOI: 10.1109/PERCOM.2018.8444577. URL: https://hal.inria.fr/hal-01704172.

[15] B. Nédelec, J. Tanke, P. Molli, A. Mostefaoui and D. Frey. 'An Adaptive Peer-Sampling Protocol for Building Networks of Browsers'. In: *World Wide Web* 25 (2017), p. 1678. DOI: 10.1007/s11280-017-0478-5. URL: https://hal.inria.fr/hal-01619906.

[16] M. Raynal and G. Taubenfeld. 'Mutual exclusion in fully anonymous shared memory systems'. In: *Information Processing Letters* 158 (June 2020), p. 105938. DOI: 10.1016/j.ipl.2020.105938. URL: https://hal.inria.fr/hal-03148640.

11.2 Publications of the year

International journals

[17] A. Auvolat, D. Frey, M. Raynal and F. Taïani. 'Money Transfer Made Simple: a Specification, a Generic Algorithm, and its Proof'. In: *Bulletin of Association for Theoretical Computer Science (BEATCS)* 132 (Oct. 2020). URL: https://hal.archives-ouvertes.fr/hal-02861511.

[18] E. Godard, D. Imbs, M. Raynal and G. Taubenfeld. 'Leader-based de-anonymization of an anonymous read/write memory'. In: *Theoretical Computer Science* 836 (Oct. 2020), pp. 110–123. DOI: 10.1016/j.tcs.2020.07.027. URL: https://hal.archives-ouvertes.fr/hal-03162641.

[19] E. Le Merrer and G. Trédan. 'Remote explainability faces the bouncer problem'. In: *Nature Machine Intelligence* 2.9 (2020), pp. 529–539. DOI: 10.1038/s42256-020-0216-z. URL: https://hal.laas.fr/hal-03048809.

[20] S. Rajsbaum and M. Raynal. '60 Years of Mastering Concurrent Computing through Sequential Thinking'. In: *ACM SIGACT News* 51.2 (16th June 2020), pp. 59–88. DOI: 10.1145/3406678.3406690. URL: https://hal.archives-ouvertes.fr/hal-03162635.

[21] M. Raynal and G. Taubenfeld. 'Mutual exclusion in fully anonymous shared memory systems'. In: *Information Processing Letters* 158 (June 2020), p. 105938. DOI: 10.1016/j.ipl.2020.105938. URL: https://hal.inria.fr/hal-03148640.

[22] K. Vargas, S. Rajsbaum and M. Raynal. 'An Eventually Perfect Failure Detector for Networks of Arbitrary Topology Connected with ADD Channels Using Time-To-Live Values'. In: *Parallel Processing Letters* 30.02 (June 2020), p. 2050006. DOI: 10.1142/S0129626420500061. URL: https://hal.archives-ouvertes.fr/hal-03162627.

[23] W. Zhu, X. Meng, X. Peng, J. Cao and M. Raynal. 'Collisions Are Preferred: RFID-Based Stocktaking with a High Missing Rate'. In: *IEEE Transactions on Mobile Computing* 19.7 (1st July 2020), pp. 1544–1554. DOI: 10.1109/TMC.2019.2911586. URL: https://hal.archives-ouvertes.fr/hal-03162646.

International peer-reviewed conferences

[24] P. Bastide, G. Giakkoupis and H. Saribekyan. 'Self-Stabilizing Clock Synchronization with 1-bit Messages'. In: ACM-SIAM Symposium on Discrete Algorithms (SODA 2021). Alexandria, VA, United States, 10th Jan. 2021. URL: https://hal.inria.fr/hal-02987598.

[25] P. Berenbrink, G. Giakkoupis and P. Kling. 'Optimal Time and Space Leader Election in Population Protocols'. In: STOC 2020 - 52nd Annual ACM Symposium on Theory of Computing. Chicago, United States, 22nd June 2020, pp. 1–29. URL: https://hal.inria.fr/hal-02545348.

[26] L. Bonniot, C. Neumann and F. Taïani. 'DiagSys: network and third-party web-service monitoring from the browser's perspective (industry track)'. In: *Middleware '20: Proceedings of the 21st International Middleware Conference Industrial Track*. 2020 - ACM/IFIP Middleware. Middleware '20: Proceedings of the 21st International Middleware Conference Industrial Track. Delft, Netherlands, 7th Dec. 2020, pp. 1–7. URL: https://hal.archives-ouvertes.fr/hal-02967290.

[27] L. Bonniot, C. Neumann and F. Taïani. 'PnyxDB: a Lightweight Leaderless Democratic Byzantine Fault Tolerant Replicated Datastore'. In: The 39th IEEE International Symposium on Reliable Distributed Systems (SRDS '20). The 39th IEEE International Symposium on Reliable Distributed Systems. Shanghai, China, 21st Sept. 2020. DOI: 10.1109/SRDS51746.2020.00023. URL: https://hal.archives-ouvertes.fr/hal-02355778.

[28] V. Cholvi, A. Fernandez Anta, C. Georgiou, N. Nicolaou and M. Raynal. 'Atomic Appends in Asynchronous Byzantine Distributed Ledgers'. In: 2020 16th European Dependable Computing Conference (EDCC). Munich, Germany, 7th Sept. 2020, pp. 77–84. DOI: 10.1109/EDCC51268.2020.00022. URL: https://hal.inria.fr/hal-03148616.

[29] F. Coulon, A. Auvolat, B. Combemale, Y.-D. Bromberg, F. Taïani, O. Barais and N. Plouzeau. 'Modular and Distributed IDE'. In: SLE 2020 - 13th ACM SIGPLAN International Conference on Software Language Engineering. Virtual, United States, Nov. 2020, pp. 270–282. DOI: 10.1145/3426425.3426947. URL: https://hal.archives-ouvertes.fr/hal-02964806.

[30] G. Damaskinos, R. Guerraoui, A.-M. Kermarrec, V. Nitu, R. Patra and F. Taïani. 'FLeet: Online Federated Learning via Staleness Awareness and Performance Prediction'. In: *Middleware '20: Proceedings of the 21st International Middleware Conference*. ACM/IFIP Middleware conference. Middleware '20: Proceedings of the 21st International Middleware Conference. Delft, Netherlands: ACM, 9th Dec. 2020. DOI: 10.1145/3423211.3425685. URL: https://hal.inria.fr/hal-03043237.

[31] G. Damaskinos, R. Guerraoui, E. Le Merrer and C. Neumann. 'The Imitation Game: Algorithm Selection by Exploiting Black-Box Recommenders'. In: NETYS 2020 - 8th International Conference on Networked Systems. NETYS. Marrakech / Virtual, Morocco, 3rd June 2020. URL: https://hal.archives-ouvertes.fr/hal-03118263.

[32] G. Giakkoupis, H. Saribekyan and T. Sauerwald. 'Spread of Information and Diseases via Random Walks in Sparse Graphs'. In: DISC 2020 - 34rd International Symposium on Distributed Computing. Freiburg, Germany, 12th Oct. 2020, pp. 1–42. URL: https://hal.inria.fr/hal-02913942.

[33] E. Godard, D. Imbs, M. Raynal and G. Taubenfeld. 'From Bezout's Identity to Space-Optimal Election in Anonymous Memory Systems'. In: PODC '20: ACM Symposium on Principles of Distributed Computing. Virtual Event Italy, Italy, 3rd Aug. 2020, pp. 41–50. DOI: 10.1145/3382734.3405727. URL: https://hal.inria.fr/hal-03148627.

[34] R. Guerraoui, A. Guirguis, A.-M. Kermarrec and E. Le Merrer. 'FeGAN: Scaling Distributed GANs'. In: ACM/IFIP Middleware 2020 - Annual ACM/IFIP Middleware conference. ACM/IFIP Middleware 2020 - Annual ACM/IFIP Middleware conference. Delft / Virtual, Netherlands, 7th Dec. 2020. URL: https://hal.archives-ouvertes.fr/hal-03118260.

[35] R. Guerraoui, A.-M. Kermarrec, O. Ruas and F. Taïani. 'Smaller, Faster & Lighter KNN Graph Constructions'. In: WWW '20 - The Web Conference 2020. Taipei Taiwan, France, 20th Apr. 2020, pp. 1060–1070. DOI: 10.1145/3366423.3380184. URL: https://hal.inria.fr/hal-02888286.

[36] A. Jaouen and E. Le Merrer. 'zoNNscan: a boundary-entropy index for zone inspection of neural models'. In: Monte Carlo Search workshop (MCS). Monte Carlos Search (MCS) workshop at IJCAI. Virtual, Japan, 12th Jan. 2021. URL: https://hal.archives-ouvertes.fr/hal-03118264.

[37] E. Le Merrer, B. Morgan and G. Trédan. 'Bug ou ban ? Une Perspective Topologique sur le Shadow Banning'. In: ALGOTEL 2020 – 22èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications. Lyon, France, 29th Sept. 2020, pp. 1–4. URL: https://hal.archives-ouvertes.fr/hal-02875595.

[38] O. Lundström, M. Raynal and E. M. Schiller. 'Self-stabilizing Uniform Reliable Broadcast'. In: NETYS. virtual online, Morocco, 14th Jan. 2021, pp. 296–313. DOI: 10.1007/978-3-030-67087-0_19. URL: https://hal.inria.fr/hal-03148662.

[39] A. B. Pilet, D. Frey and F. Taïani. 'Foiling Sybils with HAPS in Permissionless Systems: An Address-based Peer Sampling Service'. In: *2020 IEEE Symposium on Computers and Communications (ISCC)*. ISCC 2020 - IEEE Symposium on Computers and Communications. Rennes, France, 9th July 2020, pp. 1–7. DOI: 10.1109/ISCC50000.2020.9219606. URL: https://hal.archives-ouvertes.fr/hal-02965955.

Conferences without proceedings

[40] Y.-D. Bromberg and L. Gitzinger. 'DroidAutoML: A microservice architecture to automate the evaluation of Android machine learning detection systems'. In: DAIS 2020 - 20th International Conference on Distributed Applications and Interoperable Systems. Malta, Malta, 15th June 2020. URL: https://hal.archives-ouvertes.fr/hal-03146161.

[41] C. Delporte, H. Fauconnier, S. Rajsbaum and M. Raynal. 'k-Immediate Snapshot and x-Set Agreement: How Are They Related?' In: International Symposium on Stabilizing, Safety, and Security of Distributed Systems. Austin Texas (online), France, 25th Nov. 2020, pp. 97–112. DOI: 10.1007/978-3-030-64348-5_8. URL: https://hal.archives-ouvertes.fr/hal-03162658.

Scientific book chapters

[42] B. Baudry, Y.-D. Bromberg, D. Frey, A. Gómez-Boix, P. Laperdrix and F. Taïani. 'Profilage de navigateurs : état de l'art et contre-mesures'. In: *Le profilage en ligne : entre libéralisme et régulation*. 15th Oct. 2020. URL: https://hal.inria.fr/hal-03043187.

Reports & preprints

[43] A. Auvolat, Y.-D. Bromberg, D. Frey and F. Taïani. *BASALT: A Rock-Solid Foundation for Epidemic Consensus Algorithms in Very Large, Very Open Networks*. 5th Feb. 2021. URL: https://hal.inria.fr/hal-03131734.

[44] L. Bonniot, C. Neumann and F. Taïani. *DiagNet: towards a generic, Internet-scale root cause analysis solution*. 7th Apr. 2020. URL: https://hal.archives-ouvertes.fr/hal-02534888.

[45] A. Bouchra Pilet, D. Frey and F. Taïani. *Simple, Efficient and Convenient Decentralized Multi-Task Learning for Neural Networks*. 27th Nov. 2020. URL: https://hal.archives-ouvertes.fr/hal-02373338.

[46] A. Clementi, F. d'Amore, G. Giakkoupis and E. Natale. *Search via Parallel Lévy Walks on $\mathbb{Z}^2$*. Inria & Université Cote d'Azur, CNRS, I3S, Sophia Antipolis, France; Università degli Studi di Roma "Tor Vergata"; Univ Rennes, Inria, CNRS, IRISA, France, 2nd Apr. 2020. URL: https://hal.archives-ouvertes.fr/hal-02530253.

[47] G. Giakkoupis, A.-M. Kermarrec, O. Ruas and F. Taïani. *Cluster-and-Conquer: When Randomness Meets Graph Locality*. 21st Oct. 2020. URL: https://hal.inria.fr/hal-02974077.

[48] O. Ojo, A. García-Agundez, B. Girault, H. Hernández, E. Cabana, A. García-García, P. Arabshahi, C. Baquero, P. Casari, E. J. Ferreira, D. Frey, C. Georgiou, M. Goessens, A. Ishchenko, E. Jiménez, O. Kebkal, R. Lillo, R. Menezes, N. Nicolaou, A. Ortega, P. Patras, J. C. Roberts, E. Stavrakis, Y. Tanaka and A. F. Anta. *CoronaSurveys: Using Surveys with Indirect Reporting to Estimate the Incidence and Evolution of Epidemics*. 18th Feb. 2021. URL: https://hal.inria.fr/hal-03145879.

11.3 Cited publications

[49] Y. Afek and E. Gafni. 'Asynchrony from synchrony'. In: *ICDCN*. 2013, pp. 225–239.

[50] A. Ahmed and E. Ahmed. 'A survey on mobile edge computing'. In: *2016 10th International Conference on Intelligent Systems and Control (ISCO)*. Jan. 2016, pp. 1–8. DOI: 10.1109/ISCO.2016.7727082. URL: http://dx.doi.org/10.1109/ISCO.2016.7727082.

[51] T. Allard, D. Frey, G. Giakkoupis and J. Lepiller. 'Lightweight Privacy-Preserving Averaging for the Internet of Things'. In: *M4IOT 2016 - 3rd Workshop on Middleware for Context-Aware Applications in the IoT*. Trento, Italy: ACM, Dec. 2016, pp. 19–22. DOI: 10.1145/3008631.3008635. URL: https://hal.inria.fr/hal-01421986.

[52] E. Anshelevich, D. Chakrabarty, A. Hate and C. Swamy. 'Approximability of the Firefighter Problem: Computing Cuts over Time'. In: *Algorithmica* 62.1-2 (2012), pp. 520–536.

[53] D. Bernstein. 'Containers and Cloud: From LXC to Docker to Kubernetes'. In: *IEEE Cloud Computing* 1.3 (Sept. 2014), pp. 81–84. DOI: 10.1109/MCC.2014.51. URL: http://dx.doi.org/10.1109/MCC.2014.51.

[54] M. Bertier, D. Frey, R. Guerraoui, A.-M. Kermarrec and V. Leroy. 'The Gossple Anonymous Social Network'. In: *ACM/IFIP/USENIX 11th International Middleware Conference (MIDDLEWARE)*. Ed. by I. Gupta and C. Mascolo. Vol. LNCS-6452. Middleware 2010. Bangalore, India: Springer, Nov. 2010, pp. 191–211. DOI: 10.1007/978-3-642-16955-7_10. URL: https://hal.inria.fr/inria-00515693.

[55] F. Bonomi. *Connected vehicles, the internet of things, and fog computing. VANET 2011, 2011*. Keynote speech at VANET. 2011.

[56] F. Bonomi, R. Milito, J. Zhu and S. Addepalli. 'Fog Computing and Its Role in the Internet of Things'. In: *1st MCC Workshop on Mobile Cloud Computing*. 2012. DOI: 10.1145/2342509.2342513. URL: http://doi.acm.org/10.1145/2342509.2342513.

[57] A. Boutet, D. Frey, R. Guerraoui, A. Jégou and A.-M. Kermarrec. 'Privacy-Preserving Distributed Collaborative Filtering'. In: *Computing*. Special Issue on NETYS 2014 98.8 (Aug. 2016). URL: https://hal.inria.fr/hal-01251314.

[58] A. Boutet, D. Frey, R. Guerraoui, A.-M. Kermarrec and R. Patra. 'HyRec: Leveraging Browsers for Scalable Recommenders'. In: *Middleware 2014*. Bordeaux, France, Dec. 2014. DOI: 10.1145/2663165.2663315. URL: https://hal.inria.fr/hal-01080016.

[59] A. Boutet, D. Frey, R. Guerraoui, A.-M. Kermarrec, A. Rault, F. Taïani and J. Wang. 'Hide & Share: Landmark-based Similarity for Private KNN Computation'. In: *45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. Rio de Janeiro, Brazil, June 2015, pp. 263–274. DOI: 10.1109/DSN.2015.60. URL: https://hal.archives-ouvertes.fr/hal-01171492.

[60] A. Boutet, D. Frey, A. Jégou, A.-M. Kermarrec and H. Ribeiro. 'FreeRec: an Anonymous and Distributed Personalization Architecture'. In: *Computing* (Dec. 2013). URL: https://hal.inria.fr/hal-00909127.

[61] B. Cohen. *Incentives Build Robustness in BitTorrent*. 2003. URL: http://citeseer.ist.psu.edu/cohen03incentives.html.

[62] C. Delporte-Gallet, H. Fauconnier, R. Guerraoui and A. Tielmann. 'The disagreement power of an adversary'. In: *Distributed Computing* 24.3-4 (2011), pp. 137–147.

[63] A. J. Demers, D. H. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. E. Sturgis, D. C. Swinehart and D. B. Terry. 'Epidemic Algorithms for Replicated Database Maintenance'. In: *PODC*. 1987, pp. 1–12.

[64] D. Frey, R. Guerraoui, A.-M. Kermarrec, M. Monod, K. Boris, M. Martin and V. Quéma. 'Heterogeneous Gossip'. In: *Middleware 2009*. Urbana-Champaign, IL, United States, Dec. 2009. URL: https://hal.inria.fr/inria-00436125.

[65] W. M. Golab, V. Hadzilacos, D. Hendler and P. Woelfel. 'RMR-efficient implementations of comparison primitives using read and write operations'. In: *Distributed Computing* 25.2 (2012), pp. 109–162.

[66] R. Guerraoui, K. Huguenin, A.-M. Kermarrec, M. Monod and S. Prusty. 'LiFTinG: Lightweight Freerider-Tracking Protocol in Gossip'. In: *11th ACM/IFIP/USENIX International Middleware Conference (MIDDLEWARE)*. Bangalore, India, Nov. 2010. DOI: 10.1007/978-3-642-16955-7_16. URL: https://hal.inria.fr/inria-00505268.

[67] R. Guerraoui, P. Kuznetsov, M. Monti, M. Pavlovič and D.-A. Seredinschi. 'The consensus number of a cryptocurrency'. In: *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*. 2019, pp. 307–316.

[68] R. A. Holley and T. M. Liggett. 'Ergodic Theorems for Weakly Interacting Infinite Systems and the Voter Model'. In: *The Annals of Probability* 3.4 (1975), pp. 643–663.

[69] D. Imbs and M. Raynal. 'A liveness condition for concurrent objects: x-wait-freedom'. In: *Concurrency and Computation: Practice and experience* 23.17 (2011), pp. 2154–2166.

[70] F. Junqueira and K. Marzullo. 'A framework for the design of dependent-failure algorithms'. In: *Concurrency and Computation: Practice and Experience* 19.17 (2007), pp. 2255–2269.

[71] D. Kempe, J. M. Kleinberg and É. Tardos. 'Influential Nodes in a Diffusion Model for Social Networks'. In: *ICALP*. 2005, pp. 1127–1138.

[72] D. Kempe, J. M. Kleinberg and É. Tardos. 'Maximizing the Spread of Influence through a Social Network'. In: *Theory of Computing* 11 (2015), pp. 105–147.

[73] D. Kempe, J. M. Kleinberg and É. Tardos. 'Maximizing the spread of influence through a social network'. In: *KDD*. 2003, pp. 137–146.

[74] P. Kuznetsov. 'Understanding non-uniform failure models'. In: *Bulletin of the EATCS* 106 (2012), pp. 53–77.

[75] E. Lieberman, C. Hauert and M. A. Nowak. 'Evolutionary dynamics on graphs'. In: *Nature* 433.7023 (2005), pp. 312–316.

[76] M. Raynal and J. Stainer. 'Synchrony weakened by message adversaries vs asynchrony restricted by failure detectors'. In: *PODC*. Proceedings of the 2013 ACM symposium on Principles of distributed computing. Montréal, Canada: ACM, July 2013, pp. 166–175. DOI: 10.1145/2484239.2484249. URL: https://hal.inria.fr/hal-00920734.

[77] N. Santoro and P. Widmayer. 'Time is not a healer'. In: *Annual Symposium on Theoretical Aspects of Computer Science*. Springer. 1989, pp. 304–313.

[78] A. Verma, L. Pedrosa, M. Korupolu, D. Oppenheimer, E. Tune and J. Wilkes. 'Large-scale cluster management at Google with Borg'. In: *Tenth European Conference on Computer Systems (Eurosys 2015)*. ACM. 2015, p. 18.

[79] L. Zhang, F. Zhou, A. Mislove and R. Sundaram. 'Maygh: Building a CDN from Client Web Browsers'. In: *8th ACM European Conference on Computer Systems*. EuroSys '13. Prague, Czech Republic: ACM, 2013, pp. 281–294. DOI: 10.1145/2465351.2465379. URL: http://doi.acm.org/10.1145/2465351.2465379.