CEPAGE
Chercher et Essaimer dans les Plates-formes A Grande Echelle
Proposition de Projet – UR Futurs (Bordeaux)

Olivier Beaumont Nicolas Bonichon Philippe Duchon Cyril Gavoille
Nicolas Hanusse Ralf Klasing

February 5, 2007

1 Composition

- **Project Leader**
  - Olivier Beaumont, assistant professor (delegation INRIA)
- **Staff members**
  - Nicolas Bonichon, assistant professor, Univ. Bordeaux 1
  - Nicolas Hanusse, CR CNRS, LaBRI
  - Cyril Gavoille, Professor, Univ. Bordeaux 1
  - Ralf Klasing, CR CNRS, LaBRI
- **External Collaborator**
  - Philippe Duchon, assistant professor (delegation CNRS) ENSEIRB
- **Postdoctoral Students**
  - Alfredo Navarra, PostDoc (from 05/06 to 05/07), Univ. Bordeaux 1
  - Miroslaw Korzeniowski, PostDoc (from 09/06 to 09/07), INRIA
- **PhD Students**
  - Hejer Rejeb, PhD Student, Univ. Bordeaux 1, 1rst year
  - Youssou Dieng, PhD Student, Univ. Bordeaux 1, 2nd year
  - Arnaud Labourel, PhD Student, Univ. Bordeaux 1, 3rd year
- **Recent PhD Students**
  - Loris Marchal, PhD, co-superv. ENS Lyon, def. 10/06
  - Nelson Morales, PhD, co-superv. INRIA Sophia, def. 01/07

Contents

2 Overall objective

The development of interconnection networks has led to the emergence of new types of computing platforms. These platforms are characterized by heterogeneity of both processing and communication resources, geographical dispersion, and instability in terms of the number and performance of participating resources. These characteristics restrict the nature of the applications that can perform well on these platforms. Due to middleware and application deployment times, applications must be long-running and involve large amounts of data; also, only loosely-coupled applications may currently be executed on unstable platforms.

The new algorithmic challenges associated with these platforms have been approached from two different directions. On the one hand, the parallel algorithms community has largely concentrated on the problems associated with heterogeneity and large amounts of data. On the other hand, the distributed systems community has focused on scalability and fault-tolerance issues. The success of file sharing applications demonstrates the capacity of the resulting algorithms to manage huge volumes of data and users on large unstable platforms. Algorithms developed within this context are completely distributed and based on peer-to-peer (P2P for short) communication.

The goal of our project is to establish a link between these two directions, by gathering researchers from the distributed systems and parallel algorithms communities. More precisely, the objective of our project is to propose models and design efficient algorithms on large scale dynamic platforms for

- broadcasting and multicasting,
- handling queries.

We have a strong experience in designing

- scheduling algorithms (for computational tasks and collective communications),
- managing compact data structures (for routing and handling queries).

We will concentrate on the following two applications, described in detail in Section 6:

- the design of content distribution algorithms based on distributed computations of flows, randomized algorithm and network coding (with application to video streaming),
- the design of task distribution algorithms for applications consisting in collections of independent tasks sharing data (with applications to molecular dynamics).

Most of the research (including ours) currently carried out on these topics relies on a centralized knowledge of the whole (topology and performances) execution platform, whereas recent evolutions in computer networks technology yield a tremendous change in the scale of these networks. The solutions designed for scheduling and managing compact data structures must be adapted to these systems, characterized by a high dynamism of their entities (participants can join and leave at will), a potential instability of the large scale networks (on which concurrent applications are running), and the increasing probability of failure.

Peer-To-Peer systems have achieved stability and fault-tolerance, as witnessed by their wide and intensive usage, by changing the view of the networks: all communication occurs on a logical network (fixed even though resources change over time), thus abstracting the actual performance of the underlying physical network. Nevertheless, disconnecting physical and logical networks leads to low performance and a waste of resources. Moreover, due to their original use (file exchange), those systems are well suited to exact search using Distributed Hash Tables (DHT's)

and are based on fixed regular virtual topologies (Hypercubes, De Bruijn graphs...). In the context of the applications we consider, more complex queries will be required (finding the set of edges used for content distribution, finding a set of replicas covering the whole database) and, in order to reach efficiency, unstructured virtual topologies must be considered.

In this context, the main scientific challenges of our project are

- to understand the underlying physical topology and to obtain models. This requires expertise in graph theory (all the members of the project) and platform modelling (Olivier Beaumont, Nicolas Bonichon and Ralf Klasing). The obtained results will be used to focus the algorithms designed in Sections 4 and 5.

- to understand how to augment the logical topology in order to achieve the good properties of P2P systems. This requires knowledge in P2P systems and small-world networks (Olivier Beaumont, Nicolas Bonichon, Philippe Duchon, Nicolas Hanusse, Cyril Gavoille). The obtained results will be used for developing the algorithms designed in Sections 4 and 5.

- to understand how to dynamically adapt compact data structures to changes in network performance and topology (Nicolas Hanusse, Cyril Gavoille) (Section 4).

- to understand how to dynamically adapt scheduling algorithms (in particular collective communication schemes) to changes in network performance and topology, using randomized algorithms (Olivier Beaumont, Nicolas Bonichon, Nicolas Hanusse, Philippe Duchon, Ralf Klasing) (Section 5).

We will detail in Section 6 how the various expertises in the team will be employed for the considered applications.

We therefore tackle several problems related to the first of the major challenges that INRIA identified in its strategic plan (2003-2007) "Designing and mastering the future network infrastructures and communication services platforms".

3 Goal and context

3.1 General context

The recent evolutions in computer networks technology, as well as their diversification, yield a tremendous change in the use of these networks: applications and systems can now be designed at a much larger scale than before. This scaling evolution is dealing with the amount of data, the number of computers, the number of users, and the geographical diversity of these users. This race towards *large scale* computing has two major implications. First, new opportunities are offered to the applications, in particular as far as scientific computing, data bases, and file sharing are concerned. Second, a large number of parallel or distributed algorithms developed for average size systems cannot be run on large scale systems without a significant degradation of their performances. In fact, one must probably relax the constraints that the system should satisfy in order to run at a larger scale. In particular the coherence protocols designed for the distributed applications are too demanding in terms of both message and time complexity, and must therefore be adapted for running at a larger scale. Moreover, most distributed systems deployed nowadays are characterized by a high dynamism of their entities (participants can join and leave at will), a potential instability of the large scale networks (on which concurrent applications are running), and an increasing individual probability of failure. Therefore, as the size of the system increases, it becomes necessary that it adapts automatically to the changes of its components, requiring a self-organization of the system in the presence of the arrival and departure of participants, data, or resources.

As a consequence, it becomes crucial to be able to understand and model the behavior of large scale systems, for efficiently exploiting these infrastructures, in particular w.r.t. designing dedicated algorithms handling a large amount of users and/or data.

3.2 Our project

In this context, the main goal of our project is to develop algorithms for both handling queries, broadcasting and multicasting information and computations on large scale platforms. We will concentrate on point-to-point networks, where the underlying physical network can be represented by a weighted graph. These issues are related to both peer-to-peer and parallel processing applications.

On the one hand, the scientific community has learnt a lot from parallel computation, in particular w.r.t. handling heterogeneous (in terms of both computing and communication) resources. The understanding of parallel computation was achieved thanks to a fine grained modeling of the physical network and of its communication primitives. On the other hand, the tremendous success of peer-to-peer (P2P) applications for file sharing has led to the design of a large number of dedicated protocols that run in a fully distributed environment. These protocols support local decisions, and the P2P services (publication, search, node insertion, etc.) are supported by a (virtual) overlay network connecting the peers over the Internet. Up to some extent, the current P2P protocols are stable and fault-tolerant, as witnessed by their wide and intensive usage. However, the experience acquired over the last decade both in parallel computing and peer to peer systems is not sufficient to tackle new problems occurring in large scale systems.

3.3 Limitations of parallel processing solutions

In the case of parallel computation solutions, some strategies have been developed in order to cope with the intrinsic difficulty induced by resource heterogeneity. It has been proved that

changing the metric (from makespan minimization to throughput maximization) simplifies most scheduling problems, both for collective communications and parallel processing. This restricts the use of target platforms to simple and regular applications, but due to the time needed to develop and deploy applications on large scale distributed platforms, the risk of failures, the intrinsic dynamism of resources, it is unrealistic to consider tightly coupled applications involving many tight synchronization. Nevertheless, (1) it is unclear how the current models can be adapted to large scale systems, and (2) the current methodology requires the use of (at least partially) centralized subroutines that cannot be run on large scale systems. In particular, these subroutines assume the ability to gather all the information regarding the network at a single node (topology, resource performance, etc.). This assumption is unrealistic in a general purpose large size platform, in which the nodes are unstable, and whose resource characteristics can vary abruptly over time. Moreover, the proposed solutions for small to average size, stable, and dedicated environments do not satisfy the minimal requirements for self-organization and fault-tolerance, two properties that are unavoidable in a large scale context. Therefore, there is a strong need to design efficient and decentralized algorithms. This requires in particular to define new metrics adapted to large scale dynamic platforms in order to analyze the performance of the proposed algorithms.

3.4 Limitations of peer-to-peer strategies

As already noted, Peer to Peer file sharing applications have been successfully deployed on large scale dynamic platforms. Nevertheless, since our goal is the design of efficient algorithms in terms of actual performance and resource consumption, we need to concentrate on specific peer to peer environments. Indeed, P2P protocols are mostly designed for file sharing applications, and are not optimized for scientific applications, nor are they adapted to sophisticated database applications. This is mainly due to the primitive goal of designing file sharing applications, where anonymity is crucial, exact queries only are used, and all large file communications are made at IP level.

Unfortunately, the context strongly differs for the applications we consider in our project, and some of the constraints appear to be in contradiction with performance and resource consumption optimization. For instance, in these systems, due to anonymity, the number of neighboring nodes in the overlay network (i.e. the number of IP addresses known to each peer) is kept relatively low, much lower than what the memory constraints on the nodes actually impose. Such a constraint induces longer routes between peers, and is therefore in contradiction with performance. In those systems, with the main exception of the LAND overlay, the overlay network (induced by the connections of each peer) is kept as far as possible separate from the underlying physical network. This property is essential in order to cope with malicious attacks, i.e. to ensure that even if a geographic site is attacked and disconnected from the rest of the network, the overall network will remain connected. Again, since actual communications between peers occur between peers connected in the overlay network, communications between two close nodes (in the physical network) may well involve many wide area messages, and therefore such a constraint is in contradiction with performance optimization. Fortunately, in the case of file sharing applications, only queries are transmitted using the overlay network, and the communication of large files is made at IP level. On the other hand, in the case of more complex communication schemes, such as broadcast or multicast, the communication of large files is done using the overlay network, due to the lack of support, at IP level, for those complex operations. In this case, in order to achieve good results, it is crucial that virtual and physical topologies be as close as possible.

3.5 Targeted platforms

Our aim is to target large scale platforms. From parallel processing, we keep the idea that resource heterogeneity dramatically complicates scheduling problems, what imposes to restrict ourselves to simple applications (the dynamism of both the topology and the performance reinforces this constraint). We will also adopt the throughput maximization objective, though it needs to be adapted to more dynamic platforms and resources.

From previous work on peer to peer systems, we keep the idea that there is no centralized large server and that all participating nodes play a symmetric role (according to their performance in terms of memory, processing power, incoming and outgoing bandwidths, etc.), what imposes the design of self-adapting protocols, where any kind of central control should be avoided as much as possible.

Since dynamism constitutes the main difficulty in the design of algorithms on large scale dynamic platforms, we will consider several layers in dynamism:

- **Stable:** In order to establish the complexity induced by dynamism, we will first consider fully heterogeneous (in terms of both processing and communication resources) but fully stable platforms (where both topology and performance are constant over time).
- **Semi-stable:** In order to establish the complexity induced by fault-tolerance, we will then consider fully heterogeneous platforms where resource performance varies over time, but topology is fixed.
- **Unstable:** At last, we will target systems facing the arrival and departure of participants, data or resources.

3.6 Application domain, theoretical and practical validation

3.6.1 Theoretical validation

In order to analyze the performance of the proposed algorithms, we first need to define a metric adapted to the targeted platform. In particular, since resource performance and topology may change over time, the metric should also be defined from the optimal performance of the platform at any time step. For instance, if throughput maximization is concerned, the objective is to provide for the proposed algorithm an approximation ratio with respect to

$$\int_{\text{SimulationTime}} \text{OptThroughput}(t)$$

or at least

$$\min_{\text{SimulationTime}} \text{OptThroughput}(t).$$

On the other hand, in order to establish complexity and approximation results, we also need to rely on a precise theoretical model of the targeted platforms.

- At a lower level, several models have been proposed to describe interference between several simultaneous communications. In the 1-port model, a node cannot simultaneously send to (or/and receive from) more than one node. Most of the steady state scheduling results have been obtained using this model. On the other hand, some authors propose to model incoming and outgoing communication from a node using fictitious incoming and outgoing links, whose bandwidths are fixed. The main advantage of this model, although it might be slightly less accurate, is that it does not require strong synchronization and that many scheduling problems can be expressed as multi-commodity flow problems, for

which decentralized efficient algorithms are known. Another important issue is to model the bandwidth actually allocated to each communication when several communications compete for a WAN link.

- At a higher level, proving good approximation ratios on general graphs may be too difficult, and it has been observed that actual platforms often exhibit a simple structure. For instance, many real life networks satisfy small-world properties, and it has been proved, for instance, that greedy routing protocols on small world networks achieve good performance. It is therefore of interest to prove that logical (given by the interactions between hosts) and physical platforms (given by the network links) exhibit some structure in order to derive efficient algorithms.

3.6.2 General framework for validation

Low level modelling of communications

In the context of large scale dynamic platforms, it is unrealistic to determine precisely the actual topology and the contention of the underlying network at application level. Indeed, existing tools such as Alnem [LMQ03] are very much based on quasi-exhaustive determination of interferences, and it takes several days to determine the actual topology of a platform made up of a few tens of nodes. Given the dynamicity of the platforms we target, we need to rely on less sophisticated models, whose parameters can be evaluated at runtime.
Therefore, we propose to model each node by an incoming and an outgoing bandwidth and to neglect interference that appears at the heart of the network (Internet), in order to concentrate on local constraints. We are currently implementing a script, based on iperf[1] to determine the achieved bit-rates for one-to-one, one-to-many and many-to-one transfers, given the number of TCP connections, and the maximal size of the TCP windows. The next step will be to build a communication protocol that enforces a prescribed sharing of the network resources. In particular, if in the optimal solution, a node P_0 must send data at rate x_i^{OUT} to node P_i and receive data at rate y_j^{IN} from node P_j, the goal is to achieve the prescribed bitrates, provided that all capacity constraints are satisfied at each node. Our aim is to implement using Java RMI a protocol able to both evaluate the parameters of our model (incoming and outgoing bandwidths) and to ensure a prescribed sharing of communication resources.

Simulation

Once low level modelling have been obtained, it is crucial to be able to test the proposed algorithms. To do this, we will first rely on simulation rather than direct experimentation. Indeed, in order to be able to compare heuristics, it is necessary to execute those heuristics on the same platform. In particular, all changes in the topology or in the resource performance should occur at the same time during the execution of the different heuristics. In order to be able to replicate the same scenario several times, we need to rely on simulations. Moreover, the metric we have tentatively defined for providing approximation results in the case of dynamic platforms requires to compute the optimal solution at each time step, which can be done off-line if all traces for the different resources are stored. Using simulation rather than experiments can be justified if the simulator itself has been proved valid. Moreover, the modelling of communications,

[1] (`http://dast.nlanr.net/Projects/Iperf/`)

[LMQ03] A. Legrand, F. Mazoit, and M. Quinson. An application-level network mapper. Research Report RR-2003-09, LIP, ENS Lyon, France, feb 2003.

processing and their interactions may be much more complex in the simulator than in the model used to provide a theoretical approximation ratio, such as in SimGrid. In particular, sophisticated TCP models for bandwidth sharing have been implemented in SimGRID.

At a higher level, the derivation of realistic models for large scale platforms is out of the scope of our project, as explained in Section 7. Therefore, in order to obtain traces and models, we will collaborate with the GANG and ASAP projects. We already worked on these topics with the members of GANG in the ACI Pair-A-Pair (ACI Pair-A-Pair finished in 2006, but we plan to propose a follow-up, with the members of GANG and Cepage projects to ANR Blanche program). On the other hand, we also need to rely on an efficient simulator in order to test our algorithms. We have not yet chosen the discrete event simulator we will use for simulations. One attractive possibility would be to adapt SimGRID, developed in the Mescal project, to large scale dynamic environments. Indeed, a parallel version of SimGrid, based on activations is currently under development. This version will be able to deal with platforms containing more than 10^5 resources. SimGrid has been developed by Henri Casanova (U.C. San Diego) and Arnaud Legrand during his PhD (under the co-supervision of O. Beaumont).

Practical validation and scaling

Finally, we propose two main applications,

- the design of content distribution algorithms based on distributed computations of flows and network coding (with application to video streaming),
- the design of task distribution algorithms for applications consisting in "weakly" dependent tasks (tasks sharing data), with applications to molecular dynamics,

that will be described in detail in Section 6. In order to test our algorithms, we propose to implement these applications using Java RMI. The main advantages of Java RMI in our context are the ease of use and the portability. Multithreading is also a crucial feature in order to schedule concurrent communications and it does not interfere with ad-hoc routing protocols developed in the project.
The applications will first be tested on small scale platforms (using desktop workstations in the laboratory). Then, in order to test their scalability, we propose to implement them on the GRID 5000 platform.

4 Efficient queries and compact data structures

The optimization schemes for content distribution processes or for handling standard queries require a good knowledge of the physical topology or performance (latencies, throughput, ...) of the network. Assuming that some rough estimate of the physical topology is given, former theoretical results described in Section 4.1 show how to pre-process the network so that local computations are performed efficiently. Due to the dynamism of large distributed platforms, some requirements on the coding of local data structures and the udpating mechanism are needed. This last process is done using the maintenance of light virtual networks, so-called *overlay networks* (see Section 4.2). In our approach, we focus on:

- *Compression.*

  The emergence of huge distributed networks does not allow the topology of the network to be totally described at each node without any compression scheme. There are at least two reasons for this:

  - In order to guarantee that local computations are done efficiently, that is avoiding external memory requests, it may be of interest that the coding of the underlying topology can be stored within *fast memory* space. Usually, the amount of cache size is bounded.
  - The dynamism of the network implies many basic message communications to update the knowledge of each node. The smaller the message size is, the better the performance.

  The compression of any topology description should not lead to an extra cost for standard requests: distance between nodes, adjacency tests, ... Roughly speaking, a decoding process should not be necessary.

- *Routing tables.*

  Routing queries and broadcasting information on large scale platforms are tasks involving many basic message communications. The maximum performance objective imposes that basic messages are routed along paths of cost as low as possible. On the other hand, local routing decisions must be fast and the algorithms and data structures involved must support a certain amount of dynamicity in the platform (the edge costs may vary over time, and/or nodes may be inserted/deleted).

- *Local computations.*

  Although the size of the data structures is less constrained in comparison with P2P systems (due to security reasons), however, even in our collaborative framework, it is unrealistic that each node manages a complete view of the platform with the full resource characteristic. Thus, a node has to manage data structures concerning only a fraction of the whole system. In fact, a partial view of the network will be sufficient for many tasks: for instance, in order to compute the distance between two nodes (distance labeling).

- *Overlay and small world networks.*

  The processes we consider can be highly dynamic. The preprocessing usually assumed takes polynomial time. Hence, when a new process arrives, it must be dealt with in an *on-line* fashion, i.e., we do not want to totally re-compute, and the (partial) re-computation has to be simple.

In order to meet these requirements, *overlay networks* are normally implemented. These are light virtual networks, i.e., they are sparse and a local change of the physical network will only lead to a small change of the corresponding virtual network. As a result, small address books are sufficient at each node.

A specific class of overlay networks are *small-world* networks. These are efficient overlay networks for (greedy) routing tasks assuming that distance requests can be performed easily.

Of course, the main difficulty is to adapt the maintenance of local data structures to the dynamism of the network.

4.1 Compression and short data structures

4.1.1 Routing with short tables

There are several techniques to manage sub-linear size routing tables (linear in the number of nodes of the platform) while guaranteeing almost shortest paths (cf. [Gav01] for a survey of routing techniques).

Some techniques provide routes of length at most $1 + \epsilon$ times the length of the shortest one while maintaining a poly-logarithmic number of entries per routing table [AGGM05,CGMZ05,Sli05]. However, these techniques are not universal in the sense that they apply only on some class of underlying topologies. Universal schemes exist. Typically they achieve $O(\sqrt{n})$-entry local routing tables for a stretch factor of 3 in the worst case [AGM+04,TZ01]. Some experiments have shown that such methods, although universal, work very well in practice, in average, on realistic scale-free or existing topologies [KFY04].

While the fundamental question is to determine the best stretch-space trade-off for universal schemes, the challenge for platform routing would be to design specific schemes supporting reasonable dynamic changes in the topology or in the metric, and for a limited class of relevant topologies. In this direction [BLTT97] have constructed (in polynomial time) network topologies for which nodes can be labeled once such that whatever the link weights vary in time, shortest path routing tables with compacity k can be designed, i.e., for each routing table the set of

[Gav01] Cyril Gavoille. Routing in distributed networks: Overview and open problems. *ACM SIGACT News - Distributed Computing Column*, 32(1):36–52, March 2001.

[AGGM05] Ittai Abraham, Cyril Gavoille, Andrew V. Goldberg, and Dahlia Malkhi. Routing in networks with low doubling dimension. Technical Report MSR-TR-2005-175, Microsoft Research, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052 - http://www.research.microsoft.com, December 2005.

[CGMZ05] T.-H. Hubert Chan, Anupam Gupta, Bruce M. Maggs, and Shuheng Zhou. On hierarchical routing in doubling metrics. In 16^{th} *Symposium on Discrete Algorithms (SODA)*, pages 762–771. ACM-SIAM, January 2005.

[Sli05] Aleksandrs Slivkins. Distance estimation and object location via rings of neighbors. In 24^{th} *Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 41–50. ACM Press, July 2005. Appears earlier as Cornell CIS technical report TR2005-1977.

[AGM+04] Ittai Abraham, Cyril Gavoille, Dahlia Malkhi, Noam Nisan, and Mikkel Thorup. Compact name-independent routing with minimum stretch. In 16^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 20–24. ACM Press, July 2004.

[TZ01] Mikkel Thorup and Uri Zwick. Compact routing schemes. In 13^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 1–10. ACM Press, July 2001.

[KFY04] Dmitri Krioukov, Kevin Fall, and Xiaowei Yang. Compact routing on internet-like graphs. *IEEE INFOCOM*, 2004. To appear.

[BLTT97] Hans Leo Bodlaender, Jan van Leeuwen, Richard B. Tan, and Dimitrios M. Thilikos. On interval routing schemes and treewidth. *Information and Computation*, 139:92–109, November 1997.

destination using the same first outgoing edge can be grouped in at most k ranges of consecutive labels.

One other aspect of the problem would be to model a realistic typical platform topology. Natural parameters (or characteristic) for this are its low dimensionality: low Euclidean or near Euclidean networks, low growing dimension, or more generally, low doubling dimension.

4.1.2 Succinct representation of underlying topologies

In order to optimize applications the platform topology itself must be discovered, and so represented in memory with some data structures. The size of the representation is an important parameter, for instance, in order to optimize the throughput during the exploration phase of the platform.

Classical data structures for representing a graph (matrix or list) can be significantly improved when the targeted graph falls in some specific classes or obeys to some properties: the graph has bounded genus (embeddable on surface of fixed genus), bounded tree-width (or c-decomposable), or embeddabble into a bounded page number [GH05,GH99]. Typically, planar topologies with n nodes (so embeddable on the plane with no edge crossings) can by efficiently coded in linear time with at most $5n + o(n)$ bits supporting adjacency queries in constant time. This improves the classical adjacency list within a non negligible $\log n$ factor on the size (the size is about $6n \log n$ bits for edge list), and also on the query time [BGH+04,BGH03a,BGH03b].

4.1.3 Local data structures and other queries

The basic routing scheme and the overlay networks must also allow us to route other queries (other than only routing) driven by applications. Typically, divide-and-conquer parallel algorithms require to compute many nearest common ancestor (NCA) queries in some tree decomposition. In a large scale platform, if the current tree structure is fully or partially distributed, then the physical location of the NCA in the platform must be optimized. More precisely, the NCA computation must be performed from distributed pieces of information, and then addressing via the routing overlay network (cf. [AGKR04] for distributed NCA algorithms).

[GH05] Cyril Gavoille and Nicolas Hanusse. On compact encoding of pagenumber k graphs. *Discrete Mathematics & Theoretical Computer Science*, 2005. To appear.

[GH99] Cyril Gavoille and Nicolas Hanusse. Compact routing tables for graphs of bounded genus. In Jiří Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, 26^{th} *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 1644 of Lecture Notes in Computer Science, pages 351–360. Springer, July 1999.

[BGH+04] Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, Dominique Poulalhon, and Gilles Schaeffer. Planar graphs, via well-orderly maps and trees. In 30^{th} *International Workshop, Graph - Theoretic Concepts in Computer Science (WG)*, volume 3353 of Lecture Notes in Computer Science. Springer, June 2004. 270-284.

[BGH03a] Nicolas Bonichon, Cyril Gavoille, and Nicolas Hanusse. Canonical decomposition of outerplanar maps and application to enumeration, coding and generation. In 29^{th} *International Workshop, Graph - Theoretic Concepts in Computer Science (WG)*, volume 2880 of Lecture Notes in Computer Science, pages 81–92. Springer-Verlag, June 2003.

[BGH03b] Nicolas Bonichon, Cyril Gavoille, and Nicolas Hanusse. An information-theoretic upper bound of planar graphs using triangulation. In 20^{th} *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2607 of Lecture Notes in Computer Science, pages 499–510. Springer, February 2003.

[AGKR04] Stephen Alstrup, Cyril Gavoille, Haim Kaplan, and Theis Rauhe. Nearest common ancestors: A survey and a new algorithm for a distributed environment. *Theory of Computing Systems*, 37:441–456, 2004.

Recently, a theory of localized data structures has been developed (initialized by [Pel00], and see [GP03] for a survey). One associates with each node a label such that some given function (or predicate) of the node can be extracted from two or more labels. Theses labels are usually joined to the addresses or inserted into a global database index.

In relation with the project, queries involving the flow computation between any sink-target pair of a capacitated network is of great interest [KKKP04]. Dynamic labeling schemes are also available for tree models [Kor05,KP03], and need further works for their adaptation to more general topologies.

Finally, localized data structures have applications to platforms implementing large database XML file types. Roughly speaking pieces of a large XML file are distributed along some platform, and some queries (typically some SELECT ... FROM extractions) involve many tree ancestor queries [AAK+05], the XML file structure being a tree. In this framework, distributed label-based data structures avoid the storing of a huge classical index database.

4.2 Overlay and small world networks

An overlay network is a virtual network whose nodes correspond either to processors or to resources of the network. Virtual links may depend on the application; for instance, different overlay networks can be designed for routing and broadcasting.

These overlay networks should support insertion and deletion of users/resources, and thus they inherently have a high dynamism.

We should distinguish *structured* and *unstructured* overlay networks:

- In the first case, we aim at designing a network in which queries can be done efficiently: greedy routing should work well (without backtracking), the spread of a piece of information takes a very short time and few messages. The natural topology of these networks are graph of small diameter and bounded degree (De Bruijn graph for instance). However, dynamic maintenance of a precise structure is difficult and any perturbation of the topology gives no guarantee for the desired tasks.

- In the case of unstructured networks, there is no topology control. For the information retrieval task, the only attempt to bound the total number of messages consists of optimizing a flooding by taking into account statistics stored at each peer: number of requests that found an item traversing a given link, ...

In both approaches, the physical topology is not involved. To our knowledge, there exists

[Pel00] David Peleg. Informative labeling schemes for graphs. In 25^{th} *International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 1893 of Lecture Notes in Computer Science, pages 579–588. Springer, August 2000.

[GP03] Cyril Gavoille and David Peleg. Compact and localized distributed data structures. *Journal of Distributed Computing*, 16:111–120, May 2003. PODC 20-Year Special Issue.

[KKKP04] Michal Katz, Nir A. Katz, Amos Korman, and David Peleg. Labeling schemes for flow and connectivity. *SIAM Journal on Computing*, 34(1):23–40, 2004.

[Kor05] Amos Korman. General compact labeling schemes for dynamic trees. In 19^{th} *International Symposium on Distributed Computing (DISC)*, volume 3724 of Lecture Notes in Computer Science, pages 457–471. Springer, September 2005.

[KP03] Amos Korman and David Peleg. Labeling schemes for weighted dynamic tree. In 30^{th} *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2719 of Lecture Notes in Computer Science, pages 369–383. Springer, July 2003.

[AAK+05] Serge Abiteboul, Stephen Alstrup, Haim Kaplan, Tova Milo, and Theis Rauhe. Compact labeling schemes for ancestor queries. *SIAM Journal on Computing*, 2005.

only one attempt in this direction. The work of Abraham and Malhki [AMD04] deals with the design of routing tables for stable platforms.

We are interested in designing overlay topologies that take into account the physical topology. Is this goal realistic in the case of semi-stable platforms?

Another work is promising. If we relax the condition of designing an overlay network with a precise topology but with some topological properties, we might construct very efficient overlay networks. Two directions can be considered: *random graphs* and *small-world* networks.

Random graphs are promising for broadcast and have been proposed for the update of replicated databases in order to minimize the total number of messages and the time complexity [DGH+88,KSSV00]. The underlying topology is the complete graph but the communication graph (pair of nodes that effectively interact) is much more sparse. At each pulse of its local clock, each node tries to send or receive any new piece of information. The advantage of this approach is the fault-tolerance property. However, this epidemic spreading leads to a waste of messages since any node can receive many times the same update. We are interested in fixing this drawback and we think that it should be possible.

For several queries, recent solutions deal with small-world networks. This approach is inspired from experiments in social sciences [Mil67]. It suggests that adding a few (non uniform) random and uncoordinated virtual long links to every node leads to shrink drastically the diameter of the network. Moreover, paths with a small number of hops can be found [Kle00,FGP04,DHLS05].

Solutions based on network augmentation (i.e. by adding virtual links to a base network) have proved to be very promising for large scale networks. This technique is referred to as turning a network into a small-world network, also called the *small-worldization* process. Indeed, it allows to transform many arbitrary networks into networks in which search operations can be performed in a greedy fashion and very quickly (typically in time poly-logarithmic in the size of the network). This property implies that any information can be easily accessed.

Our goal is to study more precisely the algorithmic performance of these new small-world networks (w.r.t. time, memory, pertinence, fault-tolerance, auto-stabilization, ...) and to propose new networks of this kind, i.e. to construct the augmentation of the base network as well as to conceive the corresponding navigation algorithm. Like classical algorithms for routing and navigation (that are essentially based on greedy algorithms), the proposed solutions have to take into account that no entity has a global knowledge of the network. A first result in this direction is promising. In [104], we proposed an economic distributed algorithm to turn a bounded growth network into a small-world. Moreover, the practical challenge will be to adapt such constructions to dynamic networks, at least under the models that are identified as

[AMD04] Ittai Abraham, Dahlia Malkhi, and Oren Dobzinski. Land: stretch (1 + epsilon) locality-aware networks for dhts. In *Symposium of Discrete Algorithms (SODA)*, pages 550–559, 2004.

[DGH+88] Alan J. Demers, Daniel H. Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard E. Sturgis, Daniel C. Swinehart, and Douglas B. Terry. Epidemic algorithms for replicated database maintenance. *Operating Systems Review*, 22(1):8–32, 1988.

[KSSV00] Richard M. Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vöcking. Randomized rumor spreading. In *FOCS*, pages 565–574, 2000.

[Mil67] S. Milgram. The small world problem. *Psychology Today*, 61(1), 1967.

[Kle00] J. Kleinberg. The Small-World Phenomenon: An Algorithmic Perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC)*, pages 163–170, 2000.

[FGP04] P. Fraigniaud, C. Gavoille, and C. Paul. Eclecticism shrinks even small worlds. In *Proceedings of the 23rd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 169–178, 2004.

[DHLS05] P. Duchon, Nicolas HANUSSE, Emmanuelle LEBHAR, and Nicolas SCHABANEL. Could any graph be turned into a small world ? In Pierre Fraigniaud, editor, *International Symposium on Distributed Computing (DISC)*, volume 3724 of *Lecture Notes in Computer Science*, pages 511–513. Springer Verlag, 2005.

relevant.

Can the *small-worldization* process can be supported in dynamic platforms? Up to now, the literature on small-world networks only deals with the routing task. We are convinced that the small-world topologies are also relevant for other tasks: quick broadcast, search in presence of faulty nodes, ... In general, we think that maintaining a small-world topology can be much more realistic than maintaining a rigidly structured overlay network and much more efficient for several tasks in unstructured overlay networks.

5 Content distribution and independent tasks computations

5.1 Content distribution on stable platforms

5.1.1 Linear programming approach

On a large and complex platform, including cycles, using a single tree to broadcast or multicast large messages may be inefficient. An alternative consists in using several weighted broadcast trees, that will be used to transmit the different parts of a message. This approach is well-suited when the message is large and can be arbitrarily split into smaller part. In this case, the goal is to maximize the overall throughput, i.e. the maximum size of the message that can be received by all destinations per time unit once steady-state has been reached.

From a complexity point of view, this approach also proved its efficiency. Indeed, in the case of makespan minimization, even under very simple communication models, finding the best broadcast or multicast tree has been proved NP-Complete. On the other hand, when dealing with throughput maximization, finding an optimal set of weighted trees for broadcasting large messages can be achieved in polynomial time [BLMR05] (even though multicast remains NP-Complete in this context [BLMR04,BMRar]).

The sketch of the algorithm is the following. The first step consists in defining activity variables representing, for each possible destination, the fraction of the message that will transit on each edge. Then, capacity constraints for the different nodes and the different edges are gathered in a linear program. From the solution of the linear program, it is possible to build the set of weighted trees associated to the solution, together with the tight schedule of all communications. These last two steps are based on sophisticated graph theorems (fractional decomposition of a graph into trees, and fractional decomposition of a bipartite graph into matchings). It is worth noting that while this approach works for broadcasts, it does not apply to the case of multicast, which has been proved NP-Complete.

This approach has nevertheless several drawbacks in our context, since even on a stable platform, parameter evaluation may not be exact. Indeed

- the performance of all the resources must be gathered at one node, which is responsible for computing the solution of the linear program,
- the output of the whole process is a set of weighted trees and a tight schedule for communications. Neither the set of trees nor the schedule are robust against small perturbations in resource performance,
- if resource performance slightly change over time, the only possible solution to adapt to changes is to gather again all information at a given node and to recompute the optimal solution,
- multicast is NP-Complete (in fact Log-APX).

The goal of our project is to adapt existing techniques to cope with error measurements and to solve the above mentioned problems.

[BLMR05] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Pipelining broadcasts on heterogeneous platforms. *IEEE Trans. Parallel Distributed Systems*, 16(4):300–313, 2005.

[BLMR04] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Complexity results and heuristics for pipelined multicast operations on heterogeneous platforms. In *2004 International Conference on Parallel Processing (ICPP'2004)*, pages 267–274. IEEE Computer Society Press, 2004.

[BMRar] Olivier Beaumont, Loris Marchal, and Yves Robert. Complexity results for collective communications on heterogeneous platforms. *Int. Journal of High Performance Computing Applications*, 2006, to appear.

5.1.2 Network coding approach

Recently, another approach has been proposed [ACLY00,KM03,ZLG04], where the message is split into packets, and packets are combined together via linear combinations. In this framework, each packet is not routed to the set of destinations through a tree, but "information" about this packet is transmitted through linear combinations to the set of destinations.

As in the case of linear programming techniques, the first step consists in defining activity variables representing, for each possible destination, the fraction of the message that will transit on each edge. Since the communication model used for network coding is much simpler than the one we considered for linear programming techniques, this first step amounts to solving a multi-commodity flow problem. If we denote by $(P_1, \ldots, P_n)$ the different parts of the message, each node receives n linear combinations $X_j, j = 1 \ldots n$ of the P_i's. Therefore, if the transfer matrix (from the P_i's to the X_i's) is non-singular, then the node can rebuild the initial message from the different linear combinations. Determining the coefficients of the linear combinations can be done using sophisticated algebraic techniques. The determinant of the transfer matrix is expressed as a multi-variate polynomial whose coefficients are the coefficients of linear combinations. Then, it is proved that if the field where coefficients are chosen is sufficiently large, it is possible to make the choice so that the transfer matrix is non-singular.

The (huge) success of the above method is due to the fact that using coding enables to maximize throughput for multicast, whereas the problem is NP-Complete if coding is not allowed.

This approach has nevertheless several drawbacks in our context (large platforms where performances can vary over time). Indeed

- all these results have been obtained under unrealistic communication models, where nodes can perform any number of simultaneous communications,
- in order to find the solution, all resource performances have to be gathered at a single node, who is responsible for solving the multi-commodity flow problem and computing the coefficients of the linear combinations,
- all participating nodes must solve linear systems in order to decode the message. Since we are interested in throughput, the speed of this decoding process should also be taken into account, what has not been done. In the general framework, decoding is done at the destination nodes, but coding must occur at all nodes, whereas on real platforms, some of those nodes (routers, switches,...) may not have any processing capability,
- if the performances of the platform change over time, the set of linear combinations may not be valid and again, the only possible solution to adapt to changes is to gather again all information at a given node and to recompute the optimal solution.

Our goal is to modify the above mentioned techniques, in the context of our project, in order to cope with these limitations.

[ACLY00] Rudolf Ahlswede, Ning Cai, Shuo-Yen Robert Li, and Raymond W. Yeung. Network information flow. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 46(4), 2000.

[KM03] Ralf Koetter and Muriel Medard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 11(5):782–795, 2003.

[ZLG04] Y. Zhu, B. Li, and J. Guo. Multicast with network coding in application-layer overlay networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Recent Advances in Service Overlay Networks*, 22(1):107–120, 2004.

5.2 Content distribution on semi-stable platforms

Linear Programming techniques seem very difficult to adapt to environments where resource performances change over time. Fortunately, this may not be the case for the Network Coding approach, due to the following two results about decentralized multi-commodity flow algorithms and randomized Network Coding.

5.2.1 Decentralized multi-commodity flow algorithms

Awerbuch and Leighton [AL93,AL94] developed a very nice distributed algorithm for computing multi-flows (1st step of the Network Coding approach). The algorithm proposed in [AL94] consists in associating a queue to each commodity at each node for all incoming or outgoing edges. These regular queues store the flow that did not reach its destination yet. Moreover, there exists a special queue at the source of each commodity. Each queue (regular or special) is associated to a potential function, depending on the size of the queues.

The sketch of the algorithm is as follows: flow is added at sources, thus increasing the potential near the sources. The flow that reaches sinks is removed from sinks, thus decreasing the potential near the sinks. Between source and sinks, each node tries to minimize locally the potential of its queues, by moving as much flow as possible, provided that resource capacities are satisfied.

With this very simple and very natural framework, flow goes from high potential areas (the sources) to low potential areas (the sinks). This algorithm is fully decentralized since nodes make their decisions depending on their state (the size of their queues), the state of their neighbors (the size of their queues), and the capacity of neighboring links.

The remarkable property about this algorithm is that if, at any time step, the network is able to ship $(1+\epsilon)d_i$ flow units for each capacity at each time step, then the algorithm will ship at least d_i units of flow at steady state. The proof of this property is based on the overall potential of all the queues in the network, which remains bounded over time. It is worth noting that this algorithm is quasi-optimal for the metrics we defined in Section 3.6.1, since the overall throughput can be made arbitrarily close to

$$\min_{\textsc{SimulationTime}} \textsc{OptThroughput}(t).$$

Nevertheless, there is still a lot of work to do before obtaining a practical version of this algorithm and to the best of our knowledge, this algorithm has never been implemented. Indeed

- the maximal size of the queues and potential functions depends on the optimal solution of the multi-commodity flow problem (the d_i's),
- the proof of algorithm correctness is based on the remark that the overall potential remains bounded, and therefore the overall flow that did not reach its destination remains bounded. On the other hand, a positive amount of flow will never reach its destination, and will be lost. This requires an external mechanism for controlling the flow that has been lost (and will stay forever in queues),

[AL93] Baruch Awerbuch and Frank Thomson Leighton. A simple local-control approximation algorithm for multicommodity flow. In *IEEE Symposium on Foundations of Computer Science*, pages 459–468, 1993.

[AL94] Baruch Awerbuch and Tom Leighton. Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks. In *IEEE Symposium on Foundations of Computer Science*, pages 487–496, 1994.

- at last, the amount of flow transmitted on each edge at each step is a rational number. If actual messages are transmitted over the links, granularity problems must be taken into account, what has not been done to the best of our knowledge.

5.2.2 Decentralized randomized network coding algorithms

Once the solution of the underlying multi-commodity flow problem has been solved to determine the amount of data to be sent at each step on each link, network coding can be applied to decide the content of the messages that will be transmitted.

The sketch of the deterministic approach for network coding has been presented in Section 5.1.2. In this (centralized) approach, the set of linear combinations that ensure that the transfer matrix is non-singular is computed at a single node, knowing all capacity constraints.

In the decentralized randomized network coding approach, all nodes choose independently the coefficients of the linear combinations at random, and transmit the coefficients together with the message. At the end, nodes in the multicast set receive both encoded messages and the coefficients of linear combinations. As in the deterministic case, if the resulting transfer matrix is non-singular, nodes can retrieve the original message by solving a linear system.

It has been proved [CWJ03,HMS+03] that if the field where the linear combination coefficients are chosen is sufficiently large, then with high probability, the resulting matrix is non-singular.

The main advantage of the randomized approach is that, once the amount of messages to be transmitted on each edge is known, each node makes its decisions fully independently. Therefore, the algorithm is decentralized and there is no need to gather all the information about the platform at a single point.

Nevertheless, as in the previous cases, there is still a lot of work before obtaining a practical version of decentralized randomized network coding. Indeed

- in the randomized settings, the resulting transfer matrix may be singular, thus making the decoding process impossible. In this case, new linear combinations have to be sent, but this problem has not been considered yet,

- the bound on the size of the field given in [CWJ03,HMS+03] is very large when the size of the network is large, thus making the whole process impractical. Indeed, in this framework, the coefficients of the linear combinations must be sent together with the encoded message, and if the field is large, the coefficients will waste much bandwidth. This bound needs to be improved before practical implementation.

- as in the deterministic case, nodes have to decode the messages with the transfer matrix. In the deterministic case, since the transfer matrix is fixed, the inverse can be pre-computed, so that decoding amounts to a matrix vector product. In the randomized case, decoding involves the resolution of a linear system at each step. Taking this processing phase into account is crucial,

- in the randomized setting, the packets may arrive in any order, and all messages need to be stored until they can be decoded. Therefore, we face both memory (due to message storage) and latency (due to late arrival of some packets) problems, that need to be solved before considering practical implementation.

[CWJ03] P. A. Chou, Y. Wu, and K. Jain. Practical network coding. In *51st Allerton Conf. Communication, Control and Computing*. IEEE Computer Society Press, 2003.

[HMS+03] Tracey Ho, Muriel Medard, Jun Shi, Michelle Effros, and David R. Karger. On randomized network coding. In *51st Allerton Conf. Communication, Control and Computing*. IEEE Computer Society Press, 2003.

5.3 Content distribution on fully dynamic platforms

In the case of fully dynamic platforms, when both resource performances and the set of resources may change over time, we need to rely on strongly different techniques. We propose to develop content distribution algorithms based on the network model described in Section 3.6.2, where each node is represented by its incoming and its outgoing bandwidth. Most of the existing solutions for content distribution rely on a set of spanning trees [CDK+03,PP06] and the goal is to balance the load between nodes (i.e. the number of times a given node appears as a non-leaf node in the broadcast trees). We rather propose to rely on algebraic techniques developed in [DMC05] in order to avoid to maintain static data structures such as trees. More specifically, we are studying the impact of heterogeneity on this algebraic approach.

5.4 Requests and Task scheduling on semi-stable platforms

As already noted, in the context of large scale distributed unstable platforms, we need to concentrate on very simple scheduling problems (indeed, most of the scheduling problems are already NP-Complete with bad approximation ratio in the case of static homogeneous platforms when communication costs are not taken into account). Recently, many algorithms have been derived, under several communication models, for master slave tasking [BBC+04,HP04] and Divisible Load Scheduling (DLS) [BGMR96,BMR05,AGR03].

In this case, we aim at executing a large bag of independent, same-size tasks. First we assume that there is a single master, that initially holds all the (data needed for all) tasks. The problem is to determine an architecture for the execution. Which processors should the master enroll in the computation? How many tasks should be sent to each participating processor? In turn, each processor involved in the execution must decide which fraction of the tasks must be computed locally, and which fraction should be sent to which neighbor (these neighbors must be determined too).

Parallelizing the computation by spreading the execution across many processors may well be limited by the induced communication volume. Rather than aiming at makespan minimization, a more relevant objective is the optimization of the throughput in steady-state mode. There are three main reasons for focusing on the steady-state operation. First is *simplicity*, as the

[CDK+03] M. Castro, P. Druschel, A-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: High-bandwidth multicast in a cooperative environment. In *SOSP'03,Lake Bolton, New York*, 2003.

[PP06] KyoungSoo Park and Vivek S. Pai. Scale and performance in the coblitz large-file distribution service. In *In Proceedings of the Third Symposium on Networked Systems Design and Implementation (NSDI '06) San Jose, CA*, 2006.

[DMC05] S. Deb, M. Médard, and C. Choute. Algebraic gossip: A network coding approach to optimal multiple rumor mongering. *IEEE Transactions on Information Theory*, 2005.

[BBC+04] C. Banino, O. Beaumont, L. Carter, J. Ferrante, A. Legrand, and Y. Robert. Scheduling strategies for master-slave tasking on heterogeneous processor platforms. *IEEE Trans. Parallel Distributed Systems*, 15(4):319–330, 2004.

[HP04] B. Hong and V.K. Prasanna. Distributed adaptive task allocation in heterogeneous computing environments to maximize throughput. In *International Parallel and Distributed Processing Symposium IPDPS'2004*. IEEE Computer Society Press, 2004.

[BGMR96] V. Bharadwaj, D. Ghose, V. Mani, and T.G. Robertazzi. *Scheduling Divisible Loads in Parallel and Distributed Systems*. IEEE Computer Society Press, 1996.

[BMR05] Olivier Beaumont, Loris Marchal, and Yves Robert. Scheduling divisible loads with return messages on heterogeneous master-worker platforms. In *International Conference on High Performance Computing HiPC'2005*, LNCS. Springer Verlag, 2005.

[AGR03] M. Adler, Y. Gong, and A. L. Rosenberg. Optimal sharing of bags of tasks in heterogeneous clusters. In *15th ACM Symp. on Parallelism in Algorithms and Architectures (SPAA'03)*, pages 1–10. ACM Press, 2003.

steady-state scheduling is in fact a relaxation of the makespan minimization problem in which the initialization and clean-up phases are ignored. One only needs to determine, for each participating resource, which fraction of time is spent computing for which application, and which fraction of time is spent communicating with which neighbor; the actual schedule then arises naturally from these quantities.

Even if some problems remain open in the case of static platforms (especially the difficult case of return messages in the case of DLS), the problem of throughput maximization in the presence of heterogeneous processing and communication resources is now well understood. On the other hand, all proposed algorithms are based on the centralized computation of the optimal schedule and therefore, are not well suited to large scale dynamic platforms. Unstable platforms would require a totally different approach, where task replication would have to play a major role. How to choose the replication factor, and how to efficiently keep track of successfully executed copies? Another important criterion to consider are the *average* response time (or delay in the system), and *maximal* response time. In fact, designing multi-criteria algorithms capable of achieving a wide range of throughput/response time trade-offs would be very valuable.

We have to tackle the design and evaluation of distributed scheduling mechanisms. Each participating resource will take scheduling decisions based upon precise information on its immediate neighborhood. Ideally, the optimal scheduling will be reached after a series of iterations aimed at local refinements of the current solution. A first concept to investigate is that of decentralized multi-commodity flow algorithms (variants of the Awerbuch and Leighton algorithm). It may well be the case that a decentralized approach turns out to be the only realistic solution in several application contexts. However, we insist that the deployment of decentralized scheduling algorithms must be very conservative. It is mandatory to guarantee that all computations are correctly executed before aiming at optimizing them by injecting local knowledge into the scheduler.

6 Software and Practical Validation

6.1 Content distribution

The resolution of the above mentioned problems for the decentralized computations of multi-flows and network coding has many practical consequences. In particular, it provides efficient decentralized algorithms for content distribution and video streaming.

Several solutions already exist for content distribution in P2P systems. For instance, SplitStream [CDK+03] proposes to split the content across a forest of interior-node-disjoint multicast trees that distributes the forwarding load among all participating peers. To multicast the message, SplitStream constructs forests in which each peer contributes only as much forwarding bandwidth as it receives. This ensures fairness, but not throughput maximization (the performance of communication resources is not taken into account). Moreover, in a highly dynamic environment, we believe that maintaining static structures, such as trees, may have a high cost.

Other systems, such as BitTorrent [Coh03], are designed for downloading a single file in a totally connected overlay network, which we call torrent. A server is in charge of maintaining the overlay. These systems are optimized in order to incite users to share downloaded parts, what may be useless in the collaborative environment we target, but not to optimize throughput. Recently, the Avalanche [GR05] system has been proposed, which is based on network coding in order to avoid the "rare block phenomenon", that occurs in non-collaborative environments. Again, this system does not aim at maximizing performance and resource consumption.

The objective of the work on multicasting over semi-stable platforms is to produce prototypes for content distribution and video streaming. The case of video streaming is especially challenging and well-suited to the framework we propose since the natural objective is throughput maximization, and network coding may be combined with other strategies for dealing with hosts whose incoming bandwidth is too low.

The different problems to be solved in order to design this prototype are the following:

- Since efficient IP support for multicast is not yet available, the first goal is the design of an efficient overlay network. Indeed, contrarily to what happens in the case of file sharing P2P applications, the communication of the whole files will take place at application layer on the overlay network. In order to design efficient protocols, it is therefore crucial that the overlay network and the underlying physical network coincide. In the case of stable platforms, several applications have been developed in order to discover the underlying network, such as ENV [SBW99] or AlNEM [LMQ03]. Unfortunately, those tools are too slow (it takes several hours to discover the topology of a platform made of a few nodes) to be used on larger and more dynamic platforms. Epidemic processes (where nodes exchange local knowledge until convergence) may be used to achieve this goal. This requires expertise in overlay design, analysis of epidemic probabilistic processes and low level modelling of communications. Olivier Beaumont, Nicolas Bonichon, Philippe Duchon, Nicolas

[CDK+03] M. Castro, P. Druschel, A-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: High-bandwidth multicast in a cooperative environment. In *SOSP'03,Lake Bolton, New York*, 2003.

[Coh03] B. Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[GR05] C. Gkantsidis and P. Rodriguez. Network coding for large scale content distribution. *IEEE Infocom*, 2005.

[SBW99] G. Shao, F. Berman, and R. Wolski. Using effective network views to promote distributed application performance. In *International Conference on Parallel and Distributed Processing Techniques and Applications*. CSREA Press, June 1999.

[LMQ03] A. Legrand, F. Mazoit, and M. Quinson. An application-level network mapper. Research Report RR-2003-09, LIP, ENS Lyon, France, feb 2003.

Hanusse and Ralf Klasing will work on this part.

- Since the overlay is related to the underlying physical topology, it is by construction not structured. Therefore, if the nodes only know a set of very close neighbors, then the route between two distant nodes may be long in terms of number of hops. The overall latency involved for a message on such a route may therefore be much larger than the expected latency obtained using IP between these two nodes. Such a problem can be circumvented by the use of long links. Indeed, it is known that for some topologies (see Section 4.2), adding only one long link per node enables to obtain routes with poly-logarithmic expected number of hops. This requires expertise in overlay networks and small-world phenomena. Olivier Beaumont, Philippe Duchon, Nicolas Hanusse will work on this part.

- Once the overlay network is built, all the questions mentioned in Section 5.2.1 and 5.2.2 about decentralized computations of multi-flows and decentralized network coding must be solved. This requires expertise in network coding, multi-flows and randomized algorithms. Olivier Beaumont, Nicolas Bonichon and Philippe Duchon will work on this part.

- At last, for large networks, storing all data structures related to flows (from any source to any (set of) destination(s)) will not be possible, due to the huge amount of data. Therefore, techniques developed in Section 4.1 and 4.1.3 will be used to obtain compact representation of ad-hoc data structures for routing flows. This requires knowledge in compact data structures and routing. Nicolas Hanusse and Cyril Gavoille will work on this part.

6.2 Task scheduling

Another interesting scheduling problem is the case of applications sharing (large) files stored in replicated distributed databases. We deal here with a particular instance of the scheduling problem mentioned in Section 5.4. This instance involves applications that require the manipulation of large files, which are initially distributed across the platform.

It may well be the case that some files are replicated. In the target application, all tasks depend upon the whole set of files. The target platform is composed of many distant, with different computing capabilities, and which are linked through an overlay network (to be built). To each node is associated a (local) data repository. Initially, the files are stored in one or several of these repositories. We assume that a file may be duplicated, and thus simultaneously stored on several data repositories, thereby potentially speeding up the next request to access them. There may be restrictions on the possibility of duplicating the files (typically, each repository is not large enough to hold a copy of all the files). The techniques developed in Section 4.1.3 will be used to maintain dynamically efficient data structures for handling files.

Our aim is to design a prototype for both maintaining data structures and distributing files and tasks over the network.

This framework occurs for instance in the case of Monte-Carlo applications where the parameters of new simulations depend on the average behavior of the simulations previously performed. The general principle is the following: several simulations (independent tasks) are launched simultaneously with different initial parameters, and then the average behavior of these simulations is computed. Then other simulations are performed with new parameters computed from the average behavior. These parameters are tuned to ensure a much faster convergence of the method. Running such an application on a semi-stable platform is a particular instance of the scheduling problem mentioned in Section 5.4.

We will focus on a particular algorithm picked from Molecular Dynamics: calculation of Potential of Mean Force (PMF) using the technique of Adaptive Bias Force (ABF). This work

is done via a collaboration with Juan Elezgaray, IECB, Bordeaux. Here is a quick presentation of this context. Estimating the time needed for a molecule to go through a cellular membrane is an important issue in biology and medicine. Typically, the diffusion time is far too long to be computed with atomistic molecular simulations (the average time to be simulated is of order of 1s and the integration step cannot be chosen larger than 10^{-15}, due to the nature of physical interactions). Classical parallel approaches, based on domain decomposition methods, lead to very poor results due to the number of barriers. Another method to estimate this time is by calculating the PMF of the system, which is in this context the average force the molecule is subject to at a given position within or around the membrane. Recently [DP01] presented a new method, called ABF, to compute the PMF. The idea is to run a small number of simulations to estimate the PMF, and then add to the system a force that cancels the estimated PMF. With this new force, new simulations are performed starting from different configurations (distributed over the computing platform) of the system computed during the previous simulations and so on. Iterating this process, the algorithm converges quite quickly to a good estimation of the PMF with a uniform sampling along the axis of diffusion. This application has been implemented and integrated to the famous molecular dynamics software NAMD [HC04].

Our aim is to propose a distributed implementation of ABF method using NAMD. It is worth noting that NAMD is designed to run on high-end parallel platforms or clusters, but not to run efficiently on instable and distributed platforms.

The different problems to be solved in order to design this application are the following:

- Since we need to start a simulation from a valid configuration (which can represent several Mbytes) with a particular position of the molecule in the membrane, and these configurations are spread among participating nodes, we need to be able to find and to download such configuration. Therefore, the first task is to find an overlay such that those requests can be handled efficiently. The overlay may be based on the representation of the axis as an interval graph and the corresponding questions are how to build and to maintain the overlay, and how to store efficiently data structures associated to it. This requires expertise in overlay networks, compact data structures and graph theory. Olivier Beaumont, Nicolas Bonichon, Philippe Duchon, Nicolas Hanusse, Cyril Gavoille and Ralf Klasing will work on this part.

- In our context, each participating node may offer some space for storing some configurations, some bandwidth and some computing power to run simulations. The question arising here is how to distribute the simulations to nodes such that computing power of all nodes are fully used. Since nodes may join and leave the network at any time, redistributions of configurations and tasks between nodes will also be necessary (but all tasks only contribute to update the PMF, so that some tasks may fail without changing the overall result). The techniques designed for content distribution will be used to spread and redistribute the set of configurations over the set of participating nodes. This requires expertise in task scheduling and distributed storage. Olivier Beaumont, Nicolas Bonichon and Philippe Duchon will work on this part.

[DP01] E. Darve and A. Pohorille. Calculating free energies using average force. *Journal of Chemical Physics*, 115:9169–9183, 2001.

[HC04] J. Hénin and C. Chipot. Overcoming free energy barriers using unconstrained molecular dynamics simulations. *Journal of Chemical Physics*, 121:2904–2914, 2004.

7 Positioning (within INRIA)

Many INRIA teams are currently working on large scale distributed networks. In this section, we list the projects closest to our proposal and we underline the main differences between Cepage and these projects. The large number of projects working in this area is certainly due to the convergence between the parallel processing and distributed systems and distributed algorithms communities.

On the one hand, parallel platforms evolve towards the aggregation of many distributed resources (either computational or desktop grids) and therefore become more heterogeneous and less stable, what induces dramatic changes in algorithm design and scheduling strategies.

- Graal (INRIA R.A.), Mescal (INRIA R.A) and Algorille (LORIA) recently began to work on this evolution. They strongly differ from our project since they mostly concentrate on task scheduling, whereas we mostly deal with communication scheduling and queries. One notable exception is the application described in Section 6.2. Nevertheless, in the context of this application, the design of the overlay network and compact data structures are the most important issues. Moreover, we collaborate on load balancing issues with these projects in the ANR ARA Alpage project (lead by O. Beaumont). These projects mostly concentrate on small instabilities due do changes in resource performance. In this context, overlay networks are not useful since it is possible to discover the actual topology of the platform.

- Grand Large (INRIA Futurs Orsay) is also working on independent tasks distribution on large scale distributed dynamic platforms. They designed XTremWeb, a software platform for distributing such applications. Grand Large mainly concentrates on platform design and experimentation rather than the design of scheduling strategies or optimal algorithms. They also designed several strategies devoted to security issues (sandboxing, identification,...) that will be useful for the actual deployment of the applications we propose.

On the other hand, many projects originally devoted to distributed algorithms and distributed systems concentrate on large scale dynamic platforms, due to the huge success of P2P file sharing applications. These projects can be split into two main categories, depending on their main focus.

- TREC (INRIA Rocquencourt), MAESTRO (INRIA Sophia) and GANG (Inria Rocquencourt) work on the understanding of structural properties of large scale dynamic networks and the design of realistic models. This aspect is not covered in the Cepage proposal [2], but structural properties (such as low doubling dimension and small world characteristics) and actual traces of P2P networks will be used in the design of both overlay networks, compact data structures and efficient algorithms. The members of GANG also deal with the design of algorithms and applications on P2P networks. On this particular point, there are several differences between GANG and Cepage

  - We consider collaborative environments and target the design of optimal algorithms (or at least approximation algorithms) for a few well-identified applications described in Section 6 whereas GANG deals with incitation mechanisms in non-collaborative environments.
  - Most of the applications we consider in Cepage associate computations and communications. Our goal is to design overlay networks and task distribution algorithms specific to computation intensive problems.

[2] In fact, this part of the Cepage project has been removed in oder to avoid intersections with these projects.

  - We concentrate on the design and proofs for randomized algorithms, based on the specific expertise of several members of the project (Philippe Duchon and Nicolas Hanusse). More generally, we come from several computer science communities (parallel computing, distributed algorithms, routing, wireless networks, probability), what strongly influences our methodology. For instance, concerning content distribution algorithms (the main applicative intersection with GANG), we concentrate on the adaptation of static optimal algorithms to more dynamic settings, rather than the analysis of the behavior of existing content distribution algorithms.

- Regal (INRIA Rocquencourt) and ASAP (project proposal at IRISA) also deal with the design of algorithms on P2P platforms. REGAL deals with large scale replication problems for applications on dynamic distributed platforms. Members of REGAL mainly focus on the guarantee of the consistency between data replicas and on the adaptative configuration of the execution support at a low layer. Concerning the activity of Cepage, we do not concentrate on consistency but on algorithmic aspects of replication by taking into consideration trade-offs between amount of memory, load balancing and time complexity. We also share several goals with the ASAP project. The focus of ASAP is more on the design of algorithms and their practical experimentation rather than their theoretical analysis. Moreover, we focus on performance analysis of a limited number of applications (described in Section 6), graph theoretic work on overlay design and small world networks, analysis of randomized algorithms on large scale platforms and the design of compact data structures, that are not considered in the ASAP project proposal. We started collaborations with ASAP on the design and analysis of geographic overlays in the ANR ARA Alpage project.

Finally, the main originality of our project is to gather expertise in scheduling of tasks and collective communication, graph theory, design of overlay networks, compact data structures, routing and randomized algorithms. We have tried to show in Section 6 how these expertises interact and are necessary to solve the target applications considered in this project proposal.

8 Collaborations and Grants

8.1 Current and Recent International Collaboration and Grants

- **EPSRC travel grant with King's College London and the University of Liverpool**

  Travel grant, 2006-2008, on "Models and Algorithms for Scale-Free Structures", in collaboration with the Department of Computer Science, King's College London, and the Department of Computer Science, the University of Liverpool. Funded by the EPSRC. Main investigators on the UK side: Colin Cooper (King's College London) and Michele Zito (University of Liverpool). Ralf Klasing is the principal investigator on the French side.

- **Royal Society Grant with King's College London**

  Bilateral Cooperation, 2004-2006, on "Web Graphs and Web Algorithms", in collaboration with the Department of Computer Science, King's College London. Funded by the Royal Society, U.K. Main investigators on the UK side: Colin Cooper and Tomasz Radzik. Ralf Klasing is the principal investigator on the French side.

- **European COST 293 Graal**

  European COST Action: "COST 293, Graal", 2004-2008. The main objective of this COST action is to elaborate global and solid advances in the design of communication networks by letting experts and researchers with strong mathematical background meet peers specialized in communication networks, and share their mutual experience by forming a multidisciplinary scientific cooperation community. This action has more than 25 academic and 4 industrial partners from 18 European countries. (http://www.cost293.org).

- **European Cost 295 DYNAMO**

  The COST 295 is an action of the European COST program (European Cooperation in the Field of Scientific and Technical Research) inside of the Telecomunications, Information Science and Technology domain (TIST). The acronym of the COST 295 Action, is DYNAMO and stands for "Dynamic Communication Networks". The COST295 Action is motivated by the need to supply a convincing theoretical framework for the analysis and control of all modern large networks induced by the interactions between decentralized and evolving computing entities, characterized by their inherently dynamic nature. (http://cost295.net/cost295/jsp/site/Portal.jsp)

8.2 List of academic collaborators abroad

- Juraj Hromkovič (ETH Zürich, Switzerland):
  Juraj Hromkovič is the Chair of Information Technology and Education at ETH Zürich, Switzerland). We collaborate with him and his group on probabilistic and approximation methods. This collaboration if manifested by mutual visits between the research groups. Several joint papers have been published.

- Leszek Gasieniec (University of Liverpool):
  Leszek Gasieniec is the Head of the Complexity Theory and Algorithmics Group in the Department of Computer Science at the University of Liverpool. We collaborate with him and his group on graph search and graph exploration. This collaboration is manifested by mutual research visits between the research groups. In 2006, Leszek Gasieniec visited the LaBRI for one month as a guest professor. Joint papers are in preparation.

- Joseph G. Peters (Simon Fraser University, Canada):
  Joseph G. Peters is the Head of the Network Modeling Group at Simon Fraser Unversity, Burnaby, Canda. We collaborate with him in the context of modeling and algorithms for network communication. This collaboration is manifested by mutual research visits between the research groups. Several joint papers have been published. Further joint papers are in preparation.

- Walter Unger (RWTH Aachen, Germany):
  Walter Unger is a permanent member of the research group on Algorithms and Complexity at RWTH Aachen. We collaborate with him and his group on algorithmic methods for network communication. This collaboration if manifested by mutual visits between the research groups. Several joint papers have been published. Further joint papers are in preparation.

- Michele Flammini (University of L'Aquila, Italy):
  Michele Flammini is the Head of the Algorithms and Computational Complexity at the University of L'Aquila, Italy. We collaborate with him and his group on algorithmic methods and modeling of network communication. This collaboration is manifested by mutual visits between the research groups. Several joint papers have been published. Further joint papers are in preparation.

- Andrzej Pelc (Université du Québec en Outaouais, Canada):
  Andrzej Pelc is the Research Chair in Distributed Computing at the Université du Québec en Outaouais. We collaborate with him and his group on gathering and rendezvous problems in distributed networks. This collaboration is manifested by mutual research visits between the research groups. In 2006, Andrzej Pelc visited the LaBRI for one month as a guest professor. Several joint papers have been published. Further joint papers are in preparation.

- Colin Cooper, Tomasz Radzik (King's College London):
  Colin Cooper and Tomasz Radzik are permanent members of the Algorithm Design Group in the Department of Computer Science at King's College London. We have been collaborating with them within the framework of the Royal Society Grant on "Web Graphs and Web Algorithms" and the EPSRC grant on "Models and Algorithms for Scale-Free Structures". The collaboration is manifested by mutual visits between the research groups. Several joint papers have been published. Further joint papers are in preparation. Also, a joint grant application is in preparation.

- Evangelos Kranakis (Carleton University, Canada):
  Evangelos Kranakis is a professor at Carleton University. He wrote 3 books about Ad Hoc Networking, Cryptography and Computation Models. He contributed to 3 papers with Nicolas Hanusse and Danny Krizanc on mobile agents algorithmic.

- Larry Carter and Jeanne Ferrante (U.C. San Diego):
  Larry Carter and Jeanne Ferrante are professor at the University of California, San Diego. They both visited Labri for one week in 2006, and Olivier Beaumont stayed at U.C.S.D. for a total of one month since 2003. Several papers have been jointly written (IPDPS'02, IPDPS'06, IEEE TPDS) and a joint paper (IEEE TPDS) is currently under revision.

- Henri Casanova (University of Hawaii at Manoa):
  Henri Casanova is a professor at the University of Manoa. He has been working during

a stay in 2005 with Olivier Beaumont on divisible load theory (joint work published in Parallel Computing).

- David Peleg (Weizmann Institute, Israel):
  David Peleg is a specialist in distributed computing. He has wrote a book in distributed computing and has more than hundred articles in journals, and 18 papers coauthored with Gavoille. He has been Professor invited of Bordeaux University few years ago, and came several times (short visit) in LaBRI for a common 3-year bilateral project on "graph labeling" with Gavoille. His last visit was december 2006 for a Ph.D defense.

- Dahlia Malkhi, Andrew V. Goldberg, and Udi Wieder (Microsoft Research, US):
  All are coauthors of Gavoille. Recently Malkhi was PC-chair of PODC '06, the world top conference in distributed computing, and with Gavoille they were organizators of an international the workshop "LOCALITY" joint with DISC '05 conference. Malkhi is a specialist in distributed computing, and together with her student Ittai Abraham, have more than 10 papers coauthored with Gavoille. Abraham visited LaBRI in 2004, and Gavoille visited Abraham at Jerusalem University in 2005.

- Mikkel Thorup (ATT Bell Labs, US):
  Mikkel Thorup is an expert in algorithms and data structures, and has tens of STOC and FOCS papers. Recently he wrote several papers on compact routing with Uri Zwick (Tel Aviv University, IL), and one with Gavoille.

8.3 List of industrial collaborators abroad

- Cyril Banino (Yahoo, Trondheim, Norway):
  Cyril Banino did his Master degree at the University of Bordeaux in 2002 under the supervision of Olivier Beaumont and his PhD in Trondheim (N.T.N.U.). During his PhD, he worked with Olivier Beaumont on decentralized algorithms for independent tasks scheduling. This collaboration is manifested by several research visits (for a total of 5 weeks since 2003) and several joint papers (IEEE TPDS, Europar'06, IPDPS'03). He has been recently appointed at Yahoo (Trondheim), and we plan to establish a formal collaboration on document storage in large distributed databases, request scheduling and independent tasks distribution across large distributed platforms.

- Dahlia Malki (Microsoft Research Silicon Valley, California):
  Dahlia Malkhi is member of the "Distributed Systems" group and of the "Algorithms and Theory" group at Microsoft Research - Silicon Valley (MSR-SCV). In order to strenghten the already well-established collaboration with Dahlia we plan the two following actions: 1) Gavoille plan to visit MSR-SCV as consultant in a near future; and 2) to write a proposal between LaBRI and Microsoft for student exchange and funding, and in order to organize visits between members of our two teams. The themes that have been mutually selected are "Broadcasting with contents" and "Tree-likeness of the Internet network".

8.4 On the National Scene

- ANR ARA "Masse de données" Alpage (Leader: Olivier Beaumont, 2006–2009):
  Alpage focuses on the design of algorithms on large scale platforms. In particular, we will tackle the following problems

  - Large scale distributed platforms modeling

- Overlay network design
- Scheduling for regular parallel applications
- Scheduling for applications sharing large files.

The project involves the following INRIA and CNRS teams : Cepage, Graal, Mescal, Algorille, ASAP, LRI and LIX

- ACI "Masse de données" Navgraph (Leader: Nicolas Hanusse, 2003–2006):
  Navgraph is a project on data visualization based upon graph modeling. We mainly focus on applications on visual data mining for the navigation in huge graphs dedicated to video databases, genomic and topic maps.
  The project involves the following laboratories: LaBRI, LRI, LIRMM, CLIPS/IMAG, LINA, LSC, IGM

- ACI "Masse de Données" Pair à Pair (2003–2006):
  Olivier Beaumont, Philippe Duchon, Cyril Gavoille and Ralf Klasing participate in this ACI lead by Laurent Viennot (Gyroweb-GANG). The goal of this ACI is the design of peer-to-peer protocols. A follow-up of "Pair-A-Pair" is under preparation, involving all members of GANG and Cepage, and should be submitted this year to "ANR Blanche" program.

- ACI "Masse de Données" GeoComp (2004-2007):
  Cyril Gavoille and Nicolas Bonichon participates in this ACI lead by Gilles Schaeffer (LIX). GEOCOMP tackles the problem of coding geometric data structures. Members of this project propose effective solutions to do the compression almost optimally without the need of a decompression process for basic requests on the structure.

9 Teaching Activities and Scientific Responsibilities

9.1 Teaching activities

The members of CEPAGE are heavily involved in teaching activities at undergraduate level (Licence 1, 2 and 3, Master 1 and 2, Engineering Schools ENSEIRB). The teaching is carried out by members of the University as part of their teaching duties, and for CNRS (at master 2 level) as extra work. It represents more than 500 hours per year.

At master 2 level, here is a list of courses taught the last two years:

- Nicolas Hanusse
  – Graph algorithms for data visualization (2nd year MASTER "Models and Algorithms" - 2005 and 2006)
  – Distributed computing (2nd year MASTER "Models and Algorithms" - 2006)
- Cyril Gavoille
  – Introduction to Distributed Computing (2nd year MASTER "Models and Algorithms" - 2005, 2006)
  – Algorithms and Communications in Networks (2nd year MASTER "Models and Algorithms" - 2005, 2006)
  – Communication and Routing (last year of engineering school ENSEIRB 2005, 2006)
- Olivier Beaumont
  – Routing and Peer to Peer Networks (last year of engineering school ENSEIRB, 2005)
- Philippe Duchon
  – Randomized Algorithms (2nd year MASTER "Models and Algorithms" - 2006)

9.2 Program Committees (since 2003)

9.2.1 Program Chair

- HeteroPar 07 (Olivier Beaumont, chair), International Workshop on Algorithms, models, and tools for parallel computing on heterogeneous networks, Austin, 2007
- EuroPar'07 (Olivier Beaumont, Local Chair, Scheduling and Load Balancing), Rennes, France, 2007
- RenPar 06 (Olivier Beaumont, co-chair) Rencontre du Parallélisme, Perpignan, 2006
- LOCALITY '05 (Cyril Gavoile, co-chair, workshop co-located with DISC '05, sep. 26, Cracow, Poland) Locality Preserving Distributed Computing Methods
- AlgoTel '03 (co-chair, May 12-14, Banyuls-sur-mer, France) Rencontres Francophones sur les aspects Algorithmiques des Télécommunications

9.2.2 Program Committees

- Olivier Beaumont
  - IPDPS 07 IEEE International Parallel and Distributed Processing Symposium, Long Beach, USA, 2007
  - PMGC'07 Workshop on Programming Models for Grid Computing, Rio de Janeiro, Brazil, 2007

- IPDPS 06 IEEE International Parallel and Distributed Processing Symposium, Rhodes Island, Greece
- ICPADS 06 International Conference on Parallel and Distributed Systems (Minneapolis, USA, 2006)
- HeteroPar 06 International Workshop on Algorithms, models, and tools for parallel computing on heterogeneous networks (Barcelona, Spain)
- PMAA 06 International Workshop on Parallel Matrix Algorithms and Applications, Rennes, France
- IPDPS 05 IEEE International Parallel and Distributed Processing Symposium, Denver Colorado, USA
- HeteroPar 05 International Workshop on Algorithms, models, and tools for parallel computing on heterogeneous networks (Boston, Massachusetts, USA)
- RenPar 05 Rencontre du Parallélisme, Le Croisic, France
- HeteroPar 04 International Workshop on Algorithms, models, and tools for parallel computing on heterogeneous networks (Cork, Ireland)
- HeteroPar 03 International Workshop on Algorithms, models, and tools for parallel computing on heterogeneous networks (Czestochowa, Poland)
- PMAA 02 International Workshop on Parallel Matrix Algorithms and Applications, Neuchâtel, Switzerland

- Cyril Gavoille
  - DISC '07 (Sep/Oct, Lemesos, Cyprus) International Symposium on Distributed Computing
  - SPAA '07 (June 9-11, San Diego, Californie, USA) Symposium on Parallelism in Algorithms and Architectures
  - AlgoTel '07 (May 29 - Jun 1, Ile d'Oléron, France) Rencontres Francophones sur les aspects Algorithmiques des Télécommunications
  - PDCN '07 (Feb. 13-15, Innsbruck, Austria) Parallel and Distributed Computing and Networks
  - PODC '06 (July 23-26, Denver, Colorado, USA) Annual ACM Symposium on Principles of Distributed Computing
  - PDCN '06 (Feb. 14-16, Innsbruck, Austria) Parallel and Distributed Computing and Networks
  - PODC '05 (Jul. 17-20, Las Vegas, Nevada, USA) Annual ACM Symposium on Principles of Distributed Computing
  - PDCN '05 (Feb. 15-17, Innsbruck, Austria) Parallel and Distributed Computing and Networks
  - HiPC '05 (Dec. 18-21, Goa, India) International Conference On High Performance Computing
  - IWDC '05 (Dec. 27-30, Kharagpur, India) International Workshop on Distributed Computing
  - STACS '04 (Mar. 25-27, Montpellier, France) Symposium on Theoretical Aspects of Computer Science

  - SIROCCO '04 (Jun. 21-23, Smolenice, Slovakia) Colloquium on Structural Information and Communication Complexity
  - SIROCCO '03 (Jun. 18-20, Umeå, Sweden) Colloquium on Structural Information and Communication Complexity

- Nicolas Hanusse
  - ALGOTEL 06 Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Tregastel, France
  - ALGOTEL 05 Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Presqu'île de Giens, France
  - Scientific chair and main organisator of Research School "Interaction and Data Visualization", september 2004, Bordeaux

- Ralf Klasing
  - SIROCCO 06 13th Colloquium on Structural Information and Communication Complexity (2006, Chester, United Kingdom).

10 Short biographies of Cepage permanent members

- **Olivier Beaumont** received his PhD degree from the University of Rennes in 1999. From 1999 to 2001, he was assistant professor at Ecole Normale Supérieure de Lyon. Then, he was appointed at ENSEIRB in Bordeaux. In 2004, he defended his "habilitation à diriger les recherches". He is currently on leave at INRIA Futurs (délégation). His research interests focus on the design of parallel and distributed algorithms, overlay networks on large scale heterogeneous platforms and combinatorial optimization.

- **Nicolas Bonichon** received his PhD degree from the Université Bordeaux 1 in 2002. He has been holding a position as assistant professor in Université Bordeaux 1 since 2004. His research interests include distributed algorithms, compact data structure, graph drawing and enumerative combinatorics.

- **Philippe Duchon** received the PhD degree from Université Bordeaux 1 in 1998. He has been holding a position as assistant professor at ENSEIRB since 1999. His research interests range from enumerative combinatorics to distributed computing, with a focus on random models and randomized algorithms.

- **Cyril Gavoille** received the PhD degree from Ecole Normale Supérieure of Lyon in 1996. From 1996 to 2002, he was assistant professor in Bordeaux 1 University. In 2000, he defended his "habilitation à diriger les recherches" and became professor in 2002. His research interests focus on distributed compact data structures and graph algorithmic.

- **Nicolas Hanusse** received the PhD degree from Université Bordeaux 1 in 1997. As a post-doc, he spent one year at Carleton University, Canada, in 2000. In 2001, he had a position of assistant professor at LRI, Paris-Sud University. He is currently a CNRS permanent researcher in the LaBRI laboratory of Bordeaux. He is mainly interested in distributed computing and graph algorithmic.

- **Ralf Klasing** received the PhD degree from the University of Paderborn in 1995. From 1995 to 1997, he was an Assistant Professor at the University of Kiel. From 1997 to 1998, he was a Research Fellow at the University of Warwick. From 1998 to 2000, he was an Assistant Professor at RWTH Aachen. From 2000 to 2002, he was a Lecturer at King's College London. In 2002, he joined the CNRS as a permanent researcher. From 2002 to 2005, he was affiliated to the laboratory I3S in Sophia Antipolis. Currently, he is affiliated to the laboratory LaBRI in Bordeaux. His research interests include communication algorithms in networks, algorithmic methods for combinatorially hard problems, web graphs and web algorithms, optimization problems in ad-hoc wireless networks, and graph exploration.

11 Publications of project members (since 2002) in relationship with the project

References

11.1 Books and Habilitation Thesis

[1] Olivier Beaumont. *Nouvelles méthodes pour l'ordonnancement sur plates-formes hétérogènes*. PhD thesis, Habilitation à diriger des recherches de l'Université de Bordeaux 1, December 2004.

[2] Cyril Gavoille. *Structures de données compactes et distribuées*. PhD thesis, December 2000. Thèse d'habilitation à diriger les recherches, Université de Bordeaux.

[3] J. Hromkovič, R. Klasing, A. Pelc, P. Ružička, and W. Unger. *Dissemination of Information in Communication Networks: Broadcasting, Gossiping, Leader Election, and Fault-Tolerance*. Springer Monograph. Springer-Verlag, 2005.

11.2 Articles in refereed journals and book chapters

[4] Stephen Alstrup, Cyril Gavoille, Haim Kaplan, and Theis Rauhe. Nearest common ancestors: A survey and a new algorithm for a distributed environment. *Theory of Computing Systems*, 37:441–456, 2004.

[5] Cyril Banino, Olivier Beaumont, Larry Carter, Jeanne Ferrante, Arnaud Legrand, and Yves Robert. Scheduling strategies for master-slave tasking on heterogeneous processor platforms. *IEEE Trans. Parallel Distributed Systems*, 15(4):319–330, 2004.

[6] O. Beaumont, L. Marchal, and Y. Robert. Complexity results for collective communications on heterogeneous platforms. *Int. Journal of High Performance Computing Applications*, 20(1):5–17, 2006.

[7] Olivier Beaumont, Henri Casanova, Arnaud Legrand, Yves Robert, and Yang Yang. Scheduling divisible loads on star and tree networks: results and open problems. *IEEE Trans. Parallel Distributed Systems*, 16(3):207–218, 2005.

[8] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Scheduling strategies for mixed data and task parallelism on heterogeneous clusters. *Parallel Processing Letters*, 13(2), 2003.

[9] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Pipelining broadcasts on heterogeneous platforms. *IEEE Trans. Parallel Distributed Systems*, 16(4):300–313, 2005.

[10] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Steady-state scheduling on heterogeneous clusters. *Int. J. of Foundations of Computer Science*, 16(2), 2005.

[11] Olivier Beaumont, Arnaud Legrand, Fabrice Rastello, and Yves Robert. Dense linear algebra kernels on heterogeneous platforms: Redistribution issues. *Parallel Computing*, 28:155–185, 2002.

[12] Olivier Beaumont, Arnaud Legrand, and Yves Robert. Static scheduling strategies for heterogeneous systems. *Computing and Informatics*, 21:413–430, 2002.

[13] Olivier Beaumont, Arnaud Legrand, and Yves Robert. The master-slave paradigm with heterogeneous processors. *IEEE Trans. Parallel Distributed Systems*, 14(9):897–908, 2003.

[14] Olivier Beaumont, Arnaud Legrand, and Yves Robert. Scheduling divisible workloads on heterogeneous platforms. *Parallel Computing*, 29:1121–1152, 2003.

[15] Jean-Claude Bermond, Jérôme Galtier, Ralf Klasing, Nelson Morales, and Stéphane Pérennes. Hardness and approximation of gathering in static radio networks. *Parallel Processing Letters*, 16(2):165–183, June 2006.

[16] H.-J. Böckenhauer, D. Bongartz, J. Hromkovič, R. Klasing, G. Proietti, S. Seibert, and W. Unger. On the hardness of constructing minimal 2-connected spanning subgraphs in complete graphs with sharpened triangle inequality. *Theoretical Computer Science*, 326(1–3):137–153, 2004.

[17] H.-J. Böckenhauer, J. Hromkovič, R. Klasing, S. Seibert, and W. Unger. Towards the notion of stability of approximation for hard optimization tasks and the traveling salesman problem. *Theoretical Computer Science*, 285(1):3–24, 2002.

[18] N. Bonichon. A bijection between realizers of maximal plane graphs and pairs of non-crossing dyck paths. *Discrete Mathematics*, 298:104–114, 2005. FPSAC'02 Special Issue.

[19] N. Bonichon, S. Felsner, and M. Mosbah. Convex drawings of 3-connected planar graphs. *Algorithimca*, 2006. To appear.

[20] N. Bonichon and M. Mosbah. Watermelon uniform random generation with applications. *Theoretical Computer Science*, 307(2):241–256, 2003.

[21] N. Bonichon, B. Le Saëc, and M. Mosbah. Orthogonal drawings based on the stratification of planar graphs. *Discrete Mathematics*, 276(1-3):43–57, 2004. Special issue: 6th International Conference on Graph Theory - Edited by J.-L. Fouquet, I. Rusu.

[22] Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, Dominique Poulalhon, and Gilles Schaeffer. Planar graphs, via well-orderly maps and trees. *Graphs and Combinatorics*, 22(2):185–202, 2006.

[23] C. Cooper, R. Klasing, and T. Radzik. A randomized algorithm for the joining protocol in dynamic distributed networks. *Theoretical Computer Science*, 2006. To appear.

[24] C. Cooper, R. Klasing, and M. Zito. Lower bounds and algorithms for dominating sets in web graphs. *Internet Mathematics*, 2006. To appear.

[25] Yon Dourisboure, Feodor F. Dragan, Cyril Gavoille, and Chenyu Yan. Spanners for bounded tree-length graphs. *Theoretical Computer Science*, 2006. To appear.

[26] P. Duchon, Nicolas HANUSSE, Emmanuelle LEBHAR, and Nicolas SCHABANEL. Could any graph be turned into a small world? *Theoretical Computer Science*, To appear.

[27] P. Duchon, Nicolas HANUSSE, Nasser SAHEB-DJAHROMI, and Akka ZEMMARI. Broadcast in the rendezvous model. *Information and Computation*, To appear.

[28] Tamar Eilam, Cyril Gavoille, and David Peleg. Compact routing schemes with low stretch factor. *Journal of Algorithms*, 46:97–114, 2003.

[29] Michele Flammini, Ralf Klasing, Alfredo Navarra, and Stéphane Pérennes. Improved approximation results for the minimum energy broadcasting problem in wireless ad hoc networks. *Algorithmica*, 2007. to appear.

[30] Michele Flammini, Ralf Klasing, Alfredo Navarra, and Stéphane Pérennes. Tightening the upper bound for the minimum energy broadcasting. *Wireless Networks*, 2007. to appear.

[31] Pierre Fraigniaud and Cyril Gavoille. Header-size lower bounds for end-to-end communication in memoryless networks. *Computer Networks*, 2004.

[32] Pierre Fraigniaud, Cyril Gavoille, and Christophe Paul. Eclecticism shrinks even small worlds. *Journal of Distributed Computing*, 2006. To appear.

[33] Cyril Gavoille. Routing in distributed networks: Overview and open problems. *ACM SIGACT News - Distributed Computing Column*, 32(1):36–52, March 2001.

[34] CYRIL GAVOILLE and Nicolas HANUSSE. On compact encoding of pagenumber k graphs. *Discrete Mathematics & Theoretical Computer Science*, To appear. To appear.

[35] Cyril Gavoille and Martin Nehéz. Interval routing in reliability networks. *Theoretical Computer Science*, 333(3):415–432, 2005.

[36] Cyril Gavoille and Christophe Paul. Distance labeling scheme and split decomposition. *Discrete Mathematics*, 273(1-3):115–130, 2003.

[37] Cyril Gavoille and David Peleg. Compact and localized distributed data structures. *Journal of Distributed Computing*, 16:111–120, May 2003. PODC 20-Year Special Issue.

[38] Cyril Gavoille, David Peleg, Stéphane Pérennès, and Ran Raz. Distance labeling in graphs. *Journal of Algorithms*, 53(1):85–112, 2004.

[39] Cyril Gavoille and Akka Zemmari. The compactness of adaptive routing tables. *Journal of Discrete Algorithms*, 1(2):237–254, 2003.

[40] Nicolas HANUSSE, Evangelos Kranakis, and Danny Krizanc. Searching with mobile agents in networks with liars. *Discrete Applied Mathematics*, 137(1):69–85, 2004. 1st International Workshop on Combinatorics of Searching, Sorting, and Coding (COSSAC '01) (Ischia).

[41] R. Klasing and C. Laforest. Hardness results and approximation algorithms of k-tuple domination in graphs. *Information Processing Letters*, 89(2):75–83, 2004.

[42] R. Klasing, E. Markou, T. Radzik, and F. Sarracco. Approximation results for black hole search in arbitrary networks. *Theoretical Computer Science*, 2006. To appear.

[43] Ralf Klasing, Christian Laforest, Joseph G. Peters, and Nicolas Thibault. Constructing incremental sequences in graphs. *Algorithmic Operations Research*, 1(2):1–7, 2006.

[44] Ralf Klasing, Euripides Markou, Tomasz Radzik, and Fabiano Sarracco. Hardness and approximation results for black hole search in arbitrary graphs. *Theoretical Computer Science*, 2007. to appear.

11.3 Publications in Conferences and Workshops

[45] Ittai Abraham and Cyril Gavoille. Object location using path separators. In 25^{th} *Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 188–197. ACM Press, July 2006.

[46] Ittai Abraham, Cyril Gavoille, Andrew V. Goldberg, and Dahlia Malkhi. Routing in networks with low doubling dimension. In 26^{th} *International Conference on Distributed Computing Systems (ICDCS)*. IEEE Computer Society Press, July 2006.

[47] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. Routing with improved communication-space trade-off. In 18^{th} *International Symposium on Distributed Computing (DISC)*, volume 3274 of Lecture Notes in Computer Science, pages 305–319. Springer, October 2004.

[48] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. Compact routing for graphs excluding a fixed minor. In 19^{th} *International Symposium on Distributed Computing (DISC)*, volume 3724 of Lecture Notes in Computer Science, pages 442–456. Springer, September 2005.

[49] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. On space-stretch trade-offs: Lower bounds. In 18^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 217–224. ACM Press, July 2006.

[50] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. On space-stretch trade-offs: Upper bounds. In 18^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 207–216. ACM Press, July 2006.

[51] Ittai Abraham, Cyril Gavoille, Dahlia Malkhi, Noam Nisan, and Mikkel Thorup. Compact name-independent routing with minimum stretch. In 16^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 20–24. ACM Press, July 2004.

[52] Stephen Alstrup, Cyril Gavoille, Haim Kaplan, and Theis Rauhe. Nearest common ancestors: A survey and a new distributed algorithm. In 14^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 258–264. ACM Press, August 2002.

[53] C. Banino, O Beaumont, and L. Natvig. Master-slave tasking on asymmetric networks. In *Proceedings of Euro-Par 2006*, volume 4128 of *Lecture Notes in Computer Science*, pages 437–447, Dresden, Germany, August 2006. Springer.

[54] Cyril Banino, Olivier Beaumont, Arnaud Legrand, and Yves Robert. Scheduling strategies for master-slave tasking on heterogeneous processor grids. In *PARA'02: International Conference on Applied Parallel Computing*, LNCS 2367, pages 423–432. Springer Verlag, 2002.

[55] Fabrice Bazzaro and Cyril Gavoille. Localized and compact data-structure for comparability graphs. In 16^{th} *Annual International Symposium on Algorithms and Computation (ISAAC)*, volume 3827 of Lecture Notes in Computer Science, pages 1122–1131. Springer, December 2005.

[56] O. Beaumont and V. Boudet, editors. *Perpignan, Octobre 2006*. Actes de Renpar 2006, 2006.

[57] O. Beaumont, L. Carter, J. Ferrante, A. Legrand, L. Marchal, and Y. Robert. Centralized versus distributed schedulers for multiple bag-of-task applications. In *International Parallel and Distributed Processing Symposium IPDPS'2006*. IEEE Computer Society Press, 2006.

[58] O. Beaumont, E.M. Daoudi, N. Maillard, P. Manneback, and J.-L. Roch. Tradeoff to minimize extra-computations and stopping criterion tests for parallel iterative schemes. In *PMAA'04 Parallel Matrix Algorithms and Applications*. CIRM, Marseille, 2004.

[59] O. Beaumont, A.-M. Kermarrec, L. Marchal, and E Riviere. Voronet: A scalable object network based on voronoi tessellations. In *International Parallel and Distributed Processing Symposium IPDPS'2007*. IEEE Computer Society Press, 2007.

[60] O. Beaumont, L. Marchal, V. Rehn, and Y. Robert. Fifo scheduling of divisible loads with return messages under the one-port model. In *Heterogeneous Computing Workshop HCW'2006*. IEEE Computer Society Press, 2006.

[61] Olivier Beaumont, Vincent Boudet, Pierre-François Dutot, Yves Robert, and Denis Trystram. *Informatique répartie : architecture, parallélisme et système*, chapter "Fondements théoriques pour la conception d'algorithmes efficaces de gestion de ressources". Hermès Publications, 2004.

[62] Olivier Beaumont, Vincent Boudet, Arnaud Legrand, Fabrice Rastello, and Yves Robert. Static data allocation and load balancing techniques for heterogeneous systems. In C.K. Yuen, editor, *Annual Review of Scalable Computing*, volume 4, chapter 1, pages 1–37. World Scientific, 2002.

[63] Olivier Beaumont, Vincent Boudet, and Yves Robert. The iso-level scheduling heuristic for heterogeneous processors. In *PDP'2002, 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*. IEEE Computer Society Press, 2002.

[64] Olivier Beaumont, Vincent Boudet, and Yves Robert. A realistic model and an efficient heuristic for scheduling with heterogeneous processors. In *HCW'2002, the 11th Heterogeneous Computing Workshop*. IEEE Computer Society Press, 2002.

[65] Olivier Beaumont, Larry Carter, Jeanne Ferrante, Arnaud Legrand, Loris Marchal, and Yves Robert. Centralized versus distributed schedulers for multiple bag-of-task applications. In *International Parallel and Distributed Processing Symposium IPDPS'2006*. IEEE Computer Society Press, accepted for presentation, 2006.

[66] Olivier Beaumont, Larry Carter, Jeanne Ferrante, Arnaud Legrand, and Yves Robert. Bandwidth-centric allocation of independent tasks on heterogeneous platforms. In *International Parallel and Distributed Processing Symposium IPDPS'2002*. IEEE Computer Society Press, 2002.

[67] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Assessing the impact and limits of steady-state scheduling for mixed task and data parallelism on heterogeneous platforms. In *HeteroPar'2004: International Conference on Heterogeneous Computing, jointly published with ISPDC'2004: International Symposium on Parallel and Distributed Computing*. IEEE Computer Society Press, 2004.

[68] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Complexity results and heuristics for pipelined multicast operations on heterogeneous platforms. In *2004 International Conference on Parallel Processing (ICPP'2004)*, pages 267–274. IEEE Computer Society Press, 2004.

[69] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Pipelining broadcasts on heterogeneous platforms. In *IPDPS'2004*. IEEE Computer Society Press, 2004.

[70] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Steady-state scheduling on heterogeneous clusters: why and how? In *6th Workshop on Advances in Parallel and Distributed Computational Models APDCM 2004*. IEEE Computer Society Press, 2004.

[71] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Independent and divisible tasks scheduling on heterogeneous star-schaped platforms with limited memory. In *PDP'2005, 13th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pages 179–186. IEEE Computer Society Press, 2005.

[72] Olivier Beaumont, Arnaud Legrand, and Yves Robert. Ordonnancement en régime permanent pour plateformes hétérogènes. In *GRID'2002, Actes de l'école thématique sur la globalisation des ressources informatiques et des données*, pages 325–334. INRIA Lorraine, 2002.

[73] Olivier Beaumont, Arnaud Legrand, and Yves Robert. A polynomial time algorithm for allocating independent tasks on heterogeneous fork-graphs. In *ISCIS'02, 17th International Symposium on Computer and Information Sciences*. CRC Press, 2002.

[74] Olivier Beaumont, Arnaud Legrand, and Yves Robert. Static scheduling strategies for dense linear algebra kernels on heterogeneous clusters. In *Parallel Matrix Algorithms and Applications*. Université de Neuchâtel, 2002.

[75] Olivier Beaumont, Arnaud Legrand, and Yves Robert. Static scheduling strategies for heterogeneous systems. In *ISCIS XVII, Seventeenth International Symposium On Computer and Information Sciences*. CRC Press, 2002.

[76] Olivier Beaumont, Arnaud Legrand, and Yves Robert. Optimal algorithms for scheduling divisible workloads on heterogeneous systems. In *HCW'2003, the 12th Heterogeneous Computing Workshop*. IEEE Computer Society Press, 2003.

[77] Olivier Beaumont, Arnaud Legrand, and Yves Robert. Scheduling strategies for mixed data and task parallelism on heterogeneous clusters and grids. In *PDP'2003, 11th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pages 209–216. IEEE Computer Society Press, 2003.

[78] Olivier Beaumont, Loris Marchal, Veronika Rehn, and Yves Robert. Fifo scheduling of divisible loads with return messages under the one-port model. In *Heterogeneous Computing Workshop HCW'2006*. IEEE Computer Society Press, accepted for presentation, 2006.

[79] Olivier Beaumont, Loris Marchal, and Yves Robert. Broadcast trees for heterogeneous platforms. In *International Parallel and Distributed Processing Symposium IPDPS'2005*. IEEE Computer Society Press, 2005.

[80] Olivier Beaumont, Loris Marchal, and Yves Robert. Scheduling divisible loads with return messages on heterogeneous master-worker platforms. In *International Conference on High Performance Computing HiPC'2005*, LNCS. Springer Verlag, 2005.

[81] J.-C. Bermond, J. Galtier, R. Klasing, N. Morales, and S. Pérennes. Hardness and approximation of gathering in static radio networks. In *FAWN06*, Pisa, Italy, March 2006.

[82] Jean-Claude Bermond, Jérôme Galtier, Ralf Klasing, Nelson Morales, and Stéphane Pérennes. Gathering in specific radio networks. In *8èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel06), Trégastel*, May 2006.

[83] Jean-Claude Bermond, Jérôme Galtier, Ralf Klasing, Nelson Morales, and Stéphane Pérennes. Hardness and approximation of gathering in static radio networks. In *FAWN06, Pisa, Italy*, March 2006.

[84] H.-J. Böckenhauer, D. Bongartz, J. Hromkovič, R. Klasing, G. Proietti, S. Seibert, and W. Unger. On the hardness of constructing minimal 2-connected spanning subgraphs in complete graphs with sharpened triangle inequality. In *Proc. of the 22nd Conference on Foundations of Software Technology and Theoretial Computer Science (FSTTCS 2002)*, volume 2556 of *Lecture Notes in Computer Science*, pages 59–70. Springer-Verlag, 2002.

[85] H.-J. Böckenhauer, D. Bongartz, J. Hromkovič, R. Klasing, G. Proietti, S. Seibert, and W. Unger. On k-edge-connectivity problems with sharpened triangle inequality (extended abstract). In *Proc. 5th Italian Conference on Algorithms and Complexity (*CIAC 2003*)*, volume 2653 of *Lecture Notes in Computer Science*, pages 189–200. Springer-Verlag, 2003.

[86] N. Bonichon. A bijection between realizers of maximal plane graphs and pairs of non-crossing dyck paths. In *Proceedings of FPSAC'02*, pages 123–132, 2002.

[87] N. Bonichon, S. Felsner, and M. Mosbah. Convex drawings of 3-connected planar graphs - (extended abstract). In *Graph Drawing: 12th International Symposium, GD 2004*, volume 3383 of *LNCS*, pages 60–70, 2004.

[88] N. Bonichon, CYRIL GAVOILLE, and Nicolas HANUSSE. Canonical decomposition of outerplanar maps and application to enumeration, coding and generation. In 29^{th} *International Workshop, Graph - Theoretic Concepts in Computer Science (WG)*, volume 2880 of Lecture Note, pages 81–92. Springer-Verlag, 2003.

[89] N. Bonichon, CYRIL GAVOILLE, and Nicolas HANUSSE. An information-theoretic upper bound of planar graphs using triangulation. In 20^{th} *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2607 of Lecture Note, pages 499–510. Springer, 2003.

[90] N. Bonichon, CYRIL GAVOILLE, Nicolas HANUSSE, D. POULALHON, and Gilles SCHAEFFER. Planar graphs, via well-orderly maps and trees. In 30^{th} *International Workshop, Graph - Theoretic Concepts in Computer Science (WG)*, volume 3353 of Lecture Note. Springer, 2004. 270-284.

[91] N. Bonichon, B. Le Saëc, and M. Mosbah. Optimal area algorithm for planar polyline drawings. In 28^{th} *International Workshop, Graph - Theoretic Concepts in Computer Science (WG)*, volume 2573 of *LNCS*, pages 35–46. Springer-Verlag, 2002.

[92] N. Bonichon, B. Le Saëc, and M. Mosbah. Wagner's theorem on realizers. In *Proceedings of International Colloquium on Automata, Languages and Programming 2002 (ICALP'02)*, volume 2380 of *Lecture Notes in Computer Science*, pages 1043–1053. Springer-Verlag, 2002.

[93] Nicolas Bonichon, Cyril Gavoille, and Arnaud Labourel. Adjacency labeling for bounded degree trees and applications. In 6^{th} *Czech-Slovak International Symposium on Combinatorics, Graph Theory, Algorithms and Applications*, July 2006. Dedicated to Jarik Nešetřil on the occasion of his 60th birthday.

[94] Nicolas Bonichon, Cyril Gavoille, and Arnaud Labourel. Short labels by traversal and jumping. In 13^{th} *International Colloquium on Structural Information & Communication Complexity (SIROCCO)*, volume 4056 of Lecture Notes in Computer Science, pages 143–156. Springer, July 2006.

[95] C. Cooper, R. Klasing, and M. Zito. Dominating sets in web graphs. In *Proceedings of the Third Workshop on Algorithms and Models for the Web-Graph (WAW 2004)*, volume 3243 of *Lecture Notes in Computer Science*, pages 31–43. Springer-Verlag, 2004.

[96] Colin Cooper, Ralf Klasing, and Tomasz Radzik. Searching for black-hole faults in a network using multiple agents. In *Proceedings of the 10th International Conference on Principles of Distributed Systems (OPODIS 2006)*, volume 4305 of *Lecture Notes in Computer Science*, pages 320–332. Springer Verlag, December 2006.

[97] Bilel Derbel and Cyril Gavoille. Fast deterministic distributed algorithms for sparse spanners. In 13^{th} *International Colloquium on Structural Information & Communication Complexity (SIROCCO)*, volume 4056 of Lecture Notes in Computer Science, pages 100–114. Springer, July 2006.

[98] A. Don and Nicolas Hanusse. A deterministic multidimensional scaling algorithm for data visualization. In *IEEE IV2006 - International Conference on Information Visualization*, pages 511–520. IEEE, July 2006.

[99] P. Duchon, Nicolas HANUSSE, Emmanuelle LEBHAR, and Nicolas SCHABANEL. Could any graph be turned into a small-world ? In *Actes d'AlgoTel'2005 (conférence francophone sur les algorithmes de communications)*, pages –, Giens, Mai 2005.

[100] P. Duchon, Nicolas HANUSSE, Emmanuelle LEBHAR, and Nicolas SCHABANEL. Could any graph be turned into a small world ? In Pierre Fraigniaud, editor, *International Symposium on Distributed Computing (DISC)*, volume 3724 of *Lecture Notes in Computer Science*, pages 511–513. Springer Verlag, 2005.

[101] P. Duchon, Nicolas HANUSSE, Nasser SAHEB-DJAHROMI, and Akka ZEMMARI. Broadcast in the rendezvous model. In V. Diekert and M. Habib, editors, *Proceedings of STACS 2004*, volume 2996 of *Lecture Notes in Computer Science*, pages 559–570. Springer, 2004.

[102] P. Duchon, Nicolas HANUSSE, and Sébastien TIXEUIL. Optimal randomized self-stabilizing mutual exclusion on synchronous rings. In Rachid Guerraoui, editor, *Proceedings of DISC'2004*, number 3274 in Lecture Notes in Computer Science, pages 216–229, Amsterdam, October 2004. Springer.

[103] P. Duchon, Nicolas HANUSSE, and Sébastien TIXEUIL. Protocoles auto-stabilisants synchrones d'exclusion mutuelle pour les anneaux anonymes et uniformes. In *Actes d'AlgoTel 2004*, pages 135–140, Batz sur Mer, Mai 2004. Université de Rennes.

[104] Philippe Duchon, Nicolas Hanusse, Emmanuelle Lebhar, and Nicolas Schabanel. Towards small world emergence. In Uzi Vishkin, editor, *SPAA2006 - 18th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pages 225–232, PO box 11405, NY - 10286-6626, July 2006. ACM SIGACT - ACM SIGARCH, ACM Pess.

[105] M. Flammini, R. Klasing, A. Navarra, and S. Pérennes. Improved approximation results for the minimum energy broadcasting problem. In *2nd ACM/SIGMOBILE Annual International Joint Workshop on Foundation of Mobile Computing (DIALM-POMC 2004)*, pages 85–91. ACM Press, 2004.

[106] Pierre Fraigniaud and Cyril Gavoille. A space lower bound for routing in trees. In 19^{th} *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2285 of Lecture Notes in Computer Science, pages 65–75. Springer, March 2002.

[107] Pierre Fraigniaud and Cyril Gavoille. Lower bounds for oblivious single-packet end-to-end communication. In 17^{th} *International Symposium on Distributed Computing (DISC)*, volume 2848 Lecture Notes in Computer Science, pages 211–223. Springer, October 2003.

[108] Pierre Fraigniaud, Cyril Gavoille, and Christophe Paul. Eclecticism shrinks even small worlds. In 23^{rd} *Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 169–178. ACM Press, July 2004.

[109] Cyril Gavoille. Distributed data structures: A survey (invited talk). In 12^{th} *International Colloquium on Structural Information & Communication Complexity (SIROCCO)*, volume 3499 of Lecture Notes in Computer Science, page 2. Springer, May 2005.

[110] Cyril Gavoille. Distributed data structures: A survey on informative labeling schemes (invited talk). In R. Královic and P. Urzyczyn, editors, 31^{st} *International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 4162 Lecture Notes in Computer Science, page 38. Springer, August 2006.

[111] Cyril Gavoille. An overview on compact routing. In 2^{nd} *Research Workshop on Flexible Network Design*, October 2006.

[112] Cyril Gavoille and Olivier Ly. Distance labeling in hyperbolic graphs. In 16^{th} *Annual International Symposium on Algorithms and Computation (ISAAC)*, volume 3827 of Lecture Notes in Computer Science, pages 1071–1079. Springer, December 2005.

[113] Cyril Gavoille and Martin Nehéz. Interval routing in reliability networks. In 9^{th} *International Colloquium on Structural Information & Communication Complexity (SIROCCO)*, pages 149–164. Carleton University Press, June 2003.

[114] Nicolas HANUSSE, Dimitris Kavvadias, Evangelos Kranakis, and Danny Krizanc. Memoryless search algorithm in networks with faulty advice. In *IFIP-WCC'2002 (World Computer Congress); Track - International Conference on Theoretical Computer Science*, pages 206–216. Kluwer Academic, 2002. Montreal.

[115] R. Klasing, Z. Lotker, A. Navarra, and S. Pérennes. From balls and bins to points and vertices. In *Proceedings of the 16th Annual International Symposium on Algorithms and Computation (ISAAC 2005)*, volume 3827 of *Lecture Notes in Computer Science*, pages 757–766. Springer Verlag, December 2005.

[116] R. Klasing, E. Markou, T. Radzik, and F. Sarracco. Approximation bounds for black hole search problems. In *Proceedings of the 9th International Conference on Principles of Distributed Systems (OPODIS 2005)*, Lecture Notes in Computer Science. Springer Verlag, December 2005.

[117] R. Klasing, E. Markou, T. Radzik, and F. Sarracco. Hardness and approximation results for black hole search in arbitrary graphs. In *Proceedings of the 12th Colloquium on Structural Information and Communication Complexity (SIROCCO 2005)*, volume 3499 of *Lecture Notes in Computer Science*, pages 200–215. Springer Verlag, May 2005.

[118] R. Klasing, A. Navarra, A. Papadopoulos, and S. Pérennes. Adaptive broadcast consumption (abc), a new heuristic and new bounds for the minimum energy broadcast routing problem. In *Proc. 3rd FIP-TC6 Networking Conference (Networking 2004)*, volume 3042 of *Lecture Notes in Computer Science*, pages 866–877. Springer-Verlag, 2004.

[119] Ralf Klasing, Euripides Markou, and Andrzej Pelc. Gathering asynchronous oblivious mobile robots in a ring. In *Proceedings of the 17th Annual International Symposium on Algorithms and Computation (ISAAC 2006)*, volume 4288 of *Lecture Notes in Computer Science*, pages 744–753. Springer Verlag, December 2006.