Collection of Abstracts

Sébastien Tixeuil

March 29, 2007

References

[ABCDT98c] G Alari, J Beauquier, J Chacko, A K Datta, and S Tixeuil. A fault-tolerant distributed sorting algorithm on tree networks. In *Proceedings of the Seventeenth IEEE International Performance, Computing, and Communications Conference (IPCCC'98)*, 1998.

Abstract: This paper presents a self-stabilizing distributed sorting algorithm for tree networks. The distributed sorting problem can be informally described as follows: Nodes cooperate to reach a global configuration where every node, depending on its identifier, is assigned a specific final value taken from a set of input values distributed across all nodes. The input values may change in time. In our solution, the system reaches its final configuration in a finite time after the input values are stable and the faults cease. The fault-tolerance and the adaptivity to changing input is achieved using Dijkstra's paradigm of self-stabilization. A self-stabilizing algorithm, regardless of the initial system state, will converge in finite time to a set of *legitimate* states without the need for explicit exception handlers or backward recovery. Our solution is based on a continuous broadcast with acknowledgment along the tree edges to achieve the synchronization among processes in the system. It has $O(nh)$ time complexity and only $O(log(n)deg)$ memory requirement where deg is the degree of the tree and h is the height of the tree.

[ABDT98c] Luc Onana Alima, Joffroy Beauquier, Ajoy Kumar Datta, and Sébastien Tixeuil. Self-stabilization with global rooted synchronizers. In *Proceedings of the 18th International Conference on Distributed Computing Systems*, pages 102–109, Amsterdam, The Netherlands, May 1998. IEEE Press.

Abstract: We propose a self-stabilizing synchronization technique, called the *Global Rooted Synchronization*, that synchronizes processors in a tree network. This synchronizer can be used as a compiler to convert a synchronous protocol P for tree networks into a self-stabilizing version of P for tree networks. The synchronizer requires only $O(1)$ memory (other than the memory needed to maintain the tree) at each node regardless of the size of the network, stabilizes in $O(h)$ time, where h is the height of the tree, and does not invoke any global operations. Several applications of this technique are presented, such as the distributed clock synchronization and global calculus computation–each being efficient both in time and space complexity compared to the previous results.

[BDDT07j] Joffroy Beauquier, Sylvie Delaët, Shlomi Dolev, and Sébastien Tixeuil. Transient fault detectors. *Distributed Computing*, page to appear, 2007.

Abstract: In this paper we present failure detectors that detect transient failures, i.e. corruption of the system state without corrupting the program of the processors. We distinguish *task* which is the problem to solve, from *implementation* which is the algorithm that solve the problem. A task is specified as a desired output of the distributed system. The mechanism used to produce this output is not a concern of the task but a concern of the implementation. In addition we are able to classify both the *distance locality* and the *history locality* property of tasks. The distance locality is related to the diameter of the system configuration that a failure detector has to maintain in order to detect a transient fault. The history locality is related to the number of consecutive system configurations that a failure detector has to maintain in order to detect a transient fault. Both the distance and history locality of a task may give the implementation designer hints concerning the techniques and resources that are required for implementing the task.

[BDDT98c] Joffroy Beauquier, Sylvie Delaët, Shlomi Dolev, and Sébastien Tixeuil. Transient fault detectors. In Shay Kutten, editor, *Distributed Computing, 12th International Symposium, DISC '98, Andros, Greece, September 24-26, 1998, Proceedings*, pages 62–74. Springer, 1998.

Abstract: In this paper we present failure detectors that detect transient failures, i.e. corruption of the system state without corrupting the program of the processors. We distinguish *task* which is the problem to solve, from *implementation* which is the algorithm that solve the problem. A task is specified as a desired output of the distributed system. The mechanism used to produce this output is not a concern of the task but a concern of the implementation. In addition we are able to classify both the *distance locality* and the *history locality* property of tasks. The distance locality is related to the diameter of the system configuration that a failure detector has to maintain in order to detect a transient fault. The history locality is related to the number of consecutive system configurations that a failure detector has to maintain in order to detect a transient fault. Both the distance and history locality of a task may give the implementation designer hints concerning the techniques and resources that are required for implementing the task.

[BDT98r] Joffroy Beauquier, Ajoy Kumar Datta, and Sébastien Tixeuil. Self-stabilizing sorting on unidirectional rings. Technical Report 1174, Laboratoire de Recherche en Informatique, May 1998.

Abstract: We present two distributed sorting algorithms for unidirectional rings subject to transient faults. One algorithm works

using the classical store-and-forward routing scheme and the other is written for the more efficient cut-through routing scheme. The distributed sorting problem can be informally described as follows: nodes cooperate to reach a global configuration where every node, depending on its identifier, is assigned a specific output value taken from a set of input values distributed across all nodes; the input values may change in time. In our solutions, the system reaches its final correct configuration in a finite time after the input values are stable and the faults cease. The fault-tolerance and the ability to changing inputs are achieved using Dijkstra's paradigm of self-stabilization.

[BDT99c] Joffroy Beauquier, Ajoy Kumar Datta, and Sébastien Tixeuil. Self-stabilizing census with cut-through constraints. In Anish Arora, editor, *1999 ICDCS Workshop on Self-stabilizing Systems*, Austin, Texas, June 1999. IEEE Computer Society.

Abstract: We present a distributed census algorithm for unidirectional rings subject to transient faults. This algorithm is written using the efficient cut-through routing scheme, where the messages must be forwarded to a neighbor before they are completely received. The distributed census problem can be informally described as follows: The nodes cooperate to reach a global configuration where every node can determine, within finite time, which nodes are and which nodes are not present in the network. The fault-tolerance is achieved using Dijkstra's paradigm of self-stabilization. A self-stabilizing algorithm, regardless of the initial system configuration, converges in finite time to a set of *legitimate* configurations.

[BKT99r] Joffroy Beauquier, Shay Kutten, and Sébastien Tixeuil. Self-stabilization in eulerian networks with cut-through constraints. Technical Report 1200, University of Paris Sud XI, Laboratoire de Recherche en Informatique, January 1999.

Abstract: We present a distributed leader election algorithm for directed eulerian networks subject to transient faults. This algorithm is written using the efficient cut-through routing scheme, where the messages must be forwarded to a neighbor before they are completely received. The leader election problem can be informally described as follows: The nodes cooperate to reach a global configuration where a single node is elected. The fault-tolerance is achieved using Dijkstra's paradigm of self-stabilization.

[CDT02c] Yu Chen, Ajoy K Datta, and Sébastien Tixeuil. Self-stabilizing inter-domain routing in the internet. In Springer Verlag, editor, *Euro-Par'2002*, pages 749–752, Paderborn, Germany, August 2002.

Abstract: This paper reports the first self-stabilizing Border Gateway Protocol (BGP). BGP is the standard inter-domain routing protocol in the Internet. Self-stabilization is a technique to tolerate arbitrary transient faults. The routing instability in the Internet can occur due to errors in configuring the routing data structures, the routing policies, transient physical and data link problems, software bugs, and memory corruption. This instability can increase the network latency, slow down the convergence of the routing data structures, and can also cause the partitioning of networks. Most of the previous studies concentrated on routing policies to achieve the convergence of BGP while the oscillations due to transient faults were ignored. The purpose of self-stabilizing BGP is to solve the routing instability problem when this instability results from transient failures. The self-stabilizing BGP presented here provides a way to detect and automatically recover from this type of faults. Our protocol is combined with an existing protocol to make it resilient to policy conflicts as well.

[CDT02r] Yu Chen, Ajoy K. Datta, and Sébastien Tixeuil. Stabilizing inter-domain routing in the internet. Technical Report 1302, Laboratoire de Recherche en Informatique, Université Paris Sud XI, 2002.

Abstract: This paper reports the first self-stabilizing Border Gateway Protocol (BGP). BGP is the standard inter-domain routing protocol in the Internet. Self-stabilization is a technique to tolerate arbitrary transient faults. The routing instability in the Internet can occur due to errors in configuring the routing data structures, the routing policies, transient physical and data link problems, software bugs, and memory corruption. This instability can increase the network latency, slow down the convergence of the routing data structures, and can also cause the partitioning of networks. Most of the previous studies concentrated on routing policies to achieve the convergence of BGP while the oscillations due to transient faults were ignored. The purpose of self-stabilizing BGP is to solve the routing instability problem when this instability results from transient failures. The self-stabilizing BGP presented here provides a way to detect and automatically recover from this type of faults. Our protocol is combined with an existing protocol to make it resilient to policy conflicts as well.

[CDT05j] Yu Chen, Ajoy K. Datta, and Sébastien Tixeuil. Stabilizing inter-domain routing in the internet. *Journal of High Speed Networks*, 14(1):21–37, 2005.

Abstract: This paper reports the first self-stabilizing Border Gateway Protocol (BGP). BGP is the standard inter-domain routing pro-

tocol in the Internet. Self-stabilization is a technique to tolerate arbitrary transient faults. The routing instability in the Internet can occur due to errors in configuring the routing data structures, the routing policies, transient physical and data link problems, software bugs, and memory corruption. This instability can increase the network latency, slow down the convergence of the routing data structures, and can also cause the partitioning of networks. Most of the previous studies concentrated on routing policies to achieve the convergence of BGP while the oscillations due to transient faults were ignored. The purpose of self-stabilizing BGP is to solve the routing instability problem when this instability results from transient failures. The self-stabilizing BGP presented here provides a way to detect and automatically recover from this type of faults. Our protocol is combined with an existing protocol to make it resilient to policy conflicts as well.

[CHLPT06c] Michaël Cadilhac, Thomas Hérault, Richard Lassaigne, Sylvain Peyronnet, and Sébastien Tixeuil. Evaluating complex mac protocols for sensor networks with apmc. In *Proceedings of AVOCS 2006*, Nancy, September 2006.

Abstract: In this paper we present an analysis of a MAC (Medium Access Control) protocol for wireless sensor networks. The purpose of this protocol is to manage wireless media access by constructing a Time Division Media Access (TDMA) schedule. APMC (Approximate Probabilistic Model Checker) is a tool that uses approximation-based verification techniques in order to analyse the behavior of complex probabilistic systems. Using APMC, we approximately computed the probabilities of several properties of the MAC protocol being studied, thus giving some insights about it performance.

[DDLT00j] Ajoy K Datta, Jerry L Derby, James E Lawrence, and Sébastien Tixeuil. Stabilizing hierarchical routing. *Journal of Interconnexion Networks*, 1(4):283–302, 2000.

Abstract: Hierarchical routing provides a less expensive algorithm compared to the traditional all-pairs routing algorithms. We present an algorithm in this paper which benefits from the lower memory requirement, faster routing table lookup, and less costly broadcast exemplified by hierarchical routing, and yet maintains routing capability of all pairs of connected nodes even in the presence of faults, such as link/node failures and repairs, and corruption of program variables. Additionally, this algorithm solves the problem of *cluster partitioning* where nodes that are supposed to be in the same

subset of the network, become isolated due to link or node failures. Being self-stabilizing, starting from an arbitrary state (with possibly corrupted routing tables), the protocol is guaranteed to reach a configuration with routing tables containing valid entries in a finite time. The protocol automatically updates the shortest paths in the face of dynamically changing link weights. The proposed protocol also dynamically allocates/deallocates storage for the routing information as the network size changes. The algorithm works on an arbitrary topology and under a distributed daemon model.

[DDLT00r] Ajoy K. Datta, Jerry L. Derby, James E Lawrence, and Sébastien Tixeuil. Stabilizing hierarchical routing. Technical Report 1244, University of Paris Sud XI, Laboratoire de Recherche en Informatique, February 2000.

Abstract: Hierarchical routing provides a less expensive algorithm compared to the traditional all-pairs routing algorithms. We present an algorithm in this paper which benefits from the lower memory requirement, faster routing table lookup, and less costly broadcast exemplified by hierarchical routing, and yet maintains routing capability of all pairs of connected nodes even in the presence of faults, such as link/node failures and repairs, and and corruption of program variables. Additionally, this algorithm solves the problem of *cluster partitioning* where nodes that are supposed to be in the same subset of the network, become isolated due to link or node failures. Being self-stabilizing, starting from an arbitrary state (with possibly corrupted routing tables), the protocol is guaranteed to reach a configuration with routing tables containing valid entries in a finite time. The protocol automatically updates the shortest paths in the face of dynamically changing link weights. The proposed protocol also dynamically allocates/deallocates storage for the routing information as the network size changes.

[DDT03r] Sylvie Delaët, Bertrand Ducourthial, and Sébastien Tixeuil. Self-stabilization with r-operators in unreliable directed networks. Technical Report 1361, Laboratoire de Recherche en Informatique, 2003.

Abstract: This paper describes a parameterized distributed algorithm applicable to any directed network. This algorithm tolerates transient faults that corrupt the processors and communication links memory (it is self-stabilizing) as well as intermittent faults (fair loss, reorder, finite duplication of messages) on communication media. A formal proof establishes its correctness for the considered problem. The function parameter of our algorithm can be instantiated to produce distributed algorithms for both fundamental and

high level applications, such as shortest path calculus and depth-first-search tree construction. Due to fault resilience properties of our algorithm, the resulting protocols are self-stabilizing at no additional cost.

[DDT05c] Sylvie Delaët, Bertrand Ducourthial, and Sébastien Tixeuil. Self-stabilization with r-operators revisited. In *Proceedings of the Seventh Symposium on Self-stabilizing Systems (SSS'05)*, volume 3764 of *Lecture Notes in Computer Science*, pages 68–80, Barcelona, Spain, October 2005. Springer Verlag.

Abstract: We present a generic distributed algorithm for solving silents tasks such as shortest path calculus, depth-first-search tree construction, best reliable transmitters, in directed networks where communication may be only unidirectional. Our solution is written for the asynchronous message passing communication model, and tolerates multiple kinds of failures (transient and intermittent). First, our algorithm is self-stabilizing, so that it recovers correct behavior after finite time starting from an arbitrary global state caused by a transient fault. Second, it tolerates fair message loss, finite message duplication, and arbitrary message reordering, during both the stabilizing phase and the stabilized phase. This second property is most interesting since, in the context of unidirectional networks, there exists no self-stabilizing reliable data-link protocol. A formal proof establishes its correctness for the considered problem, and subsumes previous proofs for solutions in the simpler reliable shared memory communication model.

[DDT06c] Sylvie Delaët, Bertrand Ducourthial, and Sébastien Tixeuil. Self-stabilization with r-operators revisited. *Journal of Aerospace Computing, Information, and Communication*, 2006.

Abstract: We present a generic distributed algorithm for solving silents tasks such as shortest path calculus, depth-first-search tree construction, best reliable transmitters, in directed networks where communication may be only unidirectional. Our solution is written for the asynchronous message passing communication model, and tolerates multiple kinds of failures (transient and intermittent). First, our algorithm is self-stabilizing, so that it recovers correct behavior after finite time starting from an arbitrary global state caused by a transient fault. Second, it tolerates fair message loss, finite message duplication, and arbitrary message reordering, during both the stabilizing phase and the stabilized phase. This second property is most interesting since, in the context of unidirectional networks, there exists no self-stabilizing reliable data-link protocol. A

formal proof establishes its correctness for the considered problem, and subsumes previous proofs for solutions in the simpler reliable shared memory communication model.

[DDT99c] Sajal K. Das, Ajoy Kumar Datta, and Sébastien Tixeuil. Self-stabilizing algorithms in dag structured networks. In *1999 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '99)*, pages 190–197, Fremantle, Australia, June 1999. IEEE Computer Society.

Abstract: This paper describes a parameterized protocol applicable to directed acyclic graph (DAG) topologies. The function parameter of the protocol is instantiated twice to design two specific protocols: (i) the topological sorting of the successor list at every node, and (ii) a shortest path routing table construction. Both protocols are self-stabilizing and thus they are resilient to transient failures and guarantee system recovery in a finite time linear in the network diameter. From the fact that a DAG topology can be imposed on a more general topology through graph labeling protocols, the solutions presented in this paper are expected to be quite useful for a large class of distributed systems, where an optimal routing along with the robustness and fault tolerance are key factors.

[DDT99j] Sajal K. Das, Ajoy Kumar Datta, and Sébastien Tixeuil. Self-stabilizing algorithms in dag structured networks. *Parallel Processing Letters*, 9(4):563–574, December 1999.

Abstract: This paper describes a parameterized protocol applicable to directed acyclic graph (DAG) topologies. The function parameter of the protocol is instantiated twice to design two specific protocols: (i) the topological sorting of the successor list at every node, and (ii) a shortest path routing table construction. Both protocols are self-stabilizing and thus they are resilient to transient failures and guarantee system recovery in a finite time linear in the network diameter. From the fact that a DAG topology can be imposed on a more general topology through graph labeling protocols, the solutions presented in this paper are expected to be quite useful for a large class of distributed systems, where an optimal routing along with the robustness and fault tolerance are key factors.

[DGKT02c] Ajoy K Datta, Maria Gradinariu, Anthony B Kenitzki, and Sébastien Tixeuil. Self-stabilizing wormhole routing in ring networks. In *Proceedings of the IEEE International Conference on Parallel and Distributed Systems (ICPADS'2002)*, Taiwan, ROC, December 2002. Best paper award.

Abstract: Wormhole routing is most common in parallel architectures in which messages are sent in small fragments called flits. It is a lightweight and efficient method of routing messages between parallel processors. Self-stabilization is a technique that guarantees tolerance to transient faults (*e.g.* memory corruption or communication hazard) for a given protocol. Self-stabilization guarantees that the network recovers to a correct behavior in finite time, without the need for human intervention. Self-stabilization also guarantees the safety property, meaning that once the network is in a legitimate state, it will remain there until another fault occurs. This paper presents the first self-stabilizing network algorithm in the wormhole routing model, using the unidirectional ring topology. Our solution benefits from wormhole routing by providing high throughput and low latency, and from self-stabilization by ensuring automatic resilience to all possible transient failures.

[DGKT02r] Ajoy K. Datta, Maria Gradinariu, Anthony B. Kenitzki, and Sébastien Tixeuil. Self-stabilizing wormhole routing on ring networks. Technical Report 1324, Laboratoire de Recherche en Informatique, Université Paris Sud XI, 2002.

Abstract: Wormhole routing is most common in parallel architectures in which messages are sent in small fragments called flits. It is a lightweight and efficient method of routing messages between parallel processors. Self-stabilization is a technique that guarantees tolerance to transient faults (*e.g.* memory corruption or communication hazard) for a given protocol. Self-stabilization guarantees that the network recovers to a correct behavior in finite time, without the need for human intervention. Self-stabilization also guarantees the safety property, meaning that once the network is in a legitimate state, it will remain there until another fault occurs. This paper presents the first self-stabilizing network algorithm in the wormhole routing model, using the unidirectional ring topology. Our solution benefits from wormhole routing by providing high throughput and low latency, and from self-stabilization by ensuring automatic resilience to all possible transient failures.

[DGKT03j] Ajoy K. Datta, Maria Gradinariu, Anthony B. Kenitzky, and Sébastien Tixeuil. Self-stabilizing wormhole routing on ring networks. *Journal of Information Science and Engineering*, 19:401–414, 2003. Extended abstract in IEEE ICPADS 2002, best paper award.

Abstract: Wormhole routing is most common in parallel architectures in which messages are sent in small fragments called flits. It is a lightweight and efficient method of routing messages between

parallel processors. Self-stabilization is a technique that guarantees tolerance to transient faults (*e.g.* memory corruption or communication hazard) for a given protocol. Self-stabilization guarantees that the network recovers to a correct behavior in finite time, without the need for human intervention. Self-stabilization also guarantees the safety property, meaning that once the network is in a legitimate state, it will remain there until another fault occurs. This paper presents the first self-stabilizing network algorithm in the wormhole routing model, using the unidirectional ring topology. Our solution benefits from wormhole routing by providing high throughput and low latency, and from self-stabilization by ensuring automatic resilience to all possible transient failures.

[DGT00c] Ajoy K Datta, Maria Gradinariu, and Sébastien Tixeuil. Self-stabilizing mutual exclusion using unfair distributed scheduler. In *IEEE International Parallel and Distributed Processing SYmposium (IPDPS'2000)*, pages 465–470, Cancun, Mexico, May 2000. IEEE Press.

Abstract: A self-stabilizing algorithm, regardless of the initial system state, converges in finite time to a set of states that satisfy a legitimacy predicate. All existing uniform probabilistic self-stabilizing mutual exclusion algorithms designed to work under an unfair distributed scheduler suffer from the following common drawback: Once stabilized, there exists no upper bound of time between two executions of the critical section at a given processor. We present the first algorithm that guarantees such a bound ($O(n^3)$, where n is the network size) while working using an unfair distributed scheduler. Our algorithm works in an anonymous unidirectional ring of any size and has a polynomial expected stabilization time.

[DGT00r] Ajoy K. Datta, Maria Gradinariu, and Sébastien Tixeuil. Self-stabilizing mutual exclusion using unfair distributed scheduler. Technical Report 1227, University of Paris Sud XI, Laboratoire de Recherche en Informatique, February 2000.

Abstract: A self-stabilizing algorithm, regardless of the initial system state, converges in finite time to a set of states that satisfy a legitimacy predicate. All existing uniform probabilistic self-stabilizing mutual exclusion algorithms designed to work under an unfair distributed scheduler suffer from the following common drawback: Once stabilized, there exists no upper bound of time between two executions of the critical section at a given processor. We present the first algorithm that guarantees such a bound ($O(n^3)$, where n is the network size) while working using an unfair distributed scheduler.

Our algorithm works in an anonymous unidirectional ring of any size and has a polynomial expected stabilization time.

[DGT04j] Ajoy K. Datta, Maria Gradinariu, and Sébastien Tixeuil. Self-stabilizing mutual exclusion with arbitrary scheduler. *The Computer Journal*, 47(3):289–298, 2004.

Abstract: A self-stabilizing algorithm, regardless of the initial system state, converges in finite time to a set of states that satisfy a legitimacy predicate. The mutual exclusion problem is fundamental in distributed computing, since it permits processors that compete to access a shared resource to be able to synchronize and get exclusive access to the resource (*i.e.* execute their critical section). It is well known that providing self-stabilization in general uniform networks (*e.g.* anonymous rings of arbitrary size) can only be probabilistic. However, all existing uniform probabilistic self-stabilizing mutual exclusion algorithms designed to work under an unfair distributed scheduler (that may choose processors to execute their code in an arbitrary maneer) suffer from the following common drawback: Once stabilized, there exists no upper bound on time between two successive executions of the critical section at a given processor. In this paper, we present the first self-stabilizing algorithm that guarantees such a bound ($O(n^3)$, where n is the network size) while working using an unfair distributed scheduler. Our algorithm works in an anonymous unidirectional ring of any size and has a polynomial expected stabilization time.

[DGT06c] Anurag Dasgupta, Sukumar Ghosh, and Sébastien Tixeuil. Selfish stabilization. In Ajoy K. Datta and Maria Gradinariu, editors, *Eighth International Symposium on Stabilization, Safety, and Security on Distributed Systems (SSS 2006)*, Lecture Notes in Computer Science, page to appear, Dallas, Texas, November 2006. Springer Verlag.

Abstract: Stabilizing distributed systems expect all the component processes to run predefined programs that are externally mandated. In Internet scale systems, this is unrealistic, since each process may have selfish interests and motives related to maximizing its own payoff. This paper formulates the problem of selfish stabilization that shows how competition blends with cooperation in a stabilizing environment.

[DHJT02e] Sylvie Delaët, Thomas Herault, Colette Johnen, and Sébastien Tixeuil, editors. *Actes de la première journée Réseaux et Algorithmes répartis*, volume 1325, Orsay, France, June 2002. LRI, Université Paris Sud.

Abstract: These proceedings contain the papers presented at the «première journée Réseaux et Algorithmes répartis» from project STAR

[DHT04ca] Philippe Duchon, Nicolas Hanusse, and Sébastien Tixeuil. Optimal randomized self-stabilizing mutual exclusion in synchronous rings. In *Proceedings of the 18th Symposium on Distributed Computing (DISC 2004)*, number 3274 in Lecture Notes in Computer Science, pages 216–229, Amsterdam, The Nederlands, October 2004. Springer Verlag.

Abstract: We propose several self-stabilizing protocols for unidirectional, anonymous, and uniform synchronous rings of arbitrary size, where processors communicate by exchanging messages. When the size of the ring n is unknown, we better the service time by a factor of n (performing the best possible complexity for the stabilization time and the memory consumption). When the memory size is known, we present a protocol that is optimal in memory (constant and independant of n), stabilization time, and service time (both are in $\Theta(n)$).

[DHT04cb] Philippe Duchon, Nicolas Hanusse, and Sébastien Tixeuil. Protocoles autostabilisants synchrones d'exclusion mutuelle pour les anneaux anonymes. In *Proceedings of Rencontres Francophones sur l'Algorithmique des Communications (Algotel'2004)*, Batz sur Mer, France, 2004. ENST Bretagne.

Abstract: We propose several self-stabilizing protocols for unidirectional, anonymous, and uniform synchronous rings of arbitrary size, where processors communicate by exchanging messages. When the size of the ring n is unknown, we better the service time by a factor of n (performing the best possible complexity for the stabilization time and the memory consumption). When the memory size is known, we present a protocol that is optimal in memory (constant and independant of n), stabilization time, and service time (both are in $\Theta(n)$).

[DHT04r] Philippe Duchon, Nicolas Hanusse, and Sébastien Tixeuil. Optimal self-stabilizing mutual exclusion on synchronous rings. Technical Report 1389, University of Paris Sud XI, Laboratoire de Recherche en Informatique, June 2004.

Abstract: We propose several self-stabilizing protocols for unidirectional, anonymous, and uniform synchronous rings of arbitrary size, where processors communicate by exchanging messages. When the size of the ring n is unknown, we better the service time by a factor of n (performing the best possible complexity for the stabilization time and the memory consumption). When the memory size is known, we present a protocol that is optimal in memory (constant and independant of n), stabilization time, and service time (both are in $\Theta(n)$).

[DNT03ca] Sylvie Delaët, Duy-So Nguyen, and Sébastien Tixeuil. Stabilité et auto-stabilisation de bgp. In *Proceedings of Algotel 2003*, Banyuls, France, May 2003. INRIA.

Abstract: BGP (Border Gateway Protocol) est le protocole standard de routage inter-domaine dans Internet. Le routage inter-domaine permet le routage entre systèmes autonomes (tels que des Universités ou des entreprises). BGP est un protocole à vecteurs de chemins enrichi par des politiques qui permettent d'altérer les informations de routage transmises aux autres systèmes autonomes. Le problème de la stabilité du routage inter-domaine est de savoir si, en partant d'une configuration cohérente bien connue, le protocole converge vers un chemin stable vers chaque destination. Il est bien connu que ce problème est NP-complet ou NP-difficile suivant les hypothèses retenues. Plusieurs approches ont été définies pour proposer des conditions suffisantes à la stabilité d'un système. Le problème de l'auto-stabilisation du routage inter-domaine est de savoir si, en partant d'une configuration arbitraire, le protocole converge vers un chemin stable vers chaque destination. Cet article traite des relations entre les deux problèmes.

[DNT03cb] Sylvie Delaët, Duy-So Nguyen, and Sébastien Tixeuil. Stabilité et auto-stabilisation du routage inter-domaine dans internet. In *Proceedings of RIVF 2003*, pages 139–144, Hanoï, Vietnam, February 2003. Studia Informatica Universalis.

Abstract: BGP (Border Gateway Protocol) est le protocole standard de routage inter-domaine dans Internet. Le routage inter-domaine permet le routage entre systèmes autonomes (tels que des Universités ou des entreprises). BGP est un protocole à vecteurs de chemins enrichi par des politiques qui permettent d'altérer les informations de routage transmises aux autres systèmes autonomes. Le problème de la stabilité du routage inter-domaine est de savoir si, en partant d'une configuration cohérente bien connue, le protocole converge vers un chemin stable vers chaque destination. Il est bien connu que ce problème est NP-complet ou NP-difficile suivant les hypothèses retenues. Plusieurs approches ont été définies pour proposer des conditions suffisantes à la stabilité d'un système. Le problème de l'auto-stabilisation du routage inter-domaine est de savoir si, en partant d'une configuration arbitraire, le protocole converge vers un chemin stable vers chaque destination. Cet article traite des relations entre les deux problèmes.

[DNT05t] Praveen Danturi, Mikhail Nesterenko, and Sébastien Tixeuil. Self-stabilizing philosophers with generic conflicts. Technical Report TR-KSU-CS-2005-05, Kent State University, August 2005.

Abstract: We generalize the classic dining philosophers problem to allow critical section entry conflicts between non-neighbor processes. We describe a deterministic self-stabilizing solution to the new problem. We extend our solution to handle a similarly generalized drinking philosophers problem. As another extension, we describe the variant that has finite failure locality. This extension allows our algorithm to tolerate process crashes.

[DNT06c] Praveen Danturi, Mikhail Nesterenko, and Sébastien Tixeuil. Self-stabilizing philosophers with generic conflicts. In Ajoy K. Datta and Maria Gradinariu, editors, *Eighth International Symposium on Stabilization, Safety, and Security on Distributed Systems (SSS 2006)*, Lecture Notes in Computer Science, page to appear, Dallas, Texas, November 2006. Springer Verlag.

Abstract: We generalize the classic dining philosophers problem to separate the conflict and communication neighbors of each process. Communication neighbors may directly exchange information while conflict neighbors compete for the access to the exclusive critical section of code. This generalization is motivated by a number of practical problems in distributed systems including problems in wireless sensor networks. We present a self-stabilizing deterministic algorithm — KDP that solves a restricted version of the generalized problem where the conflict set for each process is limited to its k-hop neighborhood. Our algorithm is terminating. We formally prove KDP correct and evaluate its performance. We then extend KDP to handle fully generalized problem. We further extend it to handle a similarly generalized drinking philosophers problem. We describe how KDP can be implemented in wireless sensor networks and demonstrate that this implementation does not jeopardize its correctness or termination properties.

[DT00c] Sylvie Delaët and Sébastien Tixeuil. Tolerating transient and intermittent failures. In *International Conference on Principles of Distributed Systems (OPODIS'2000)*, pages 17–36, Paris, France, December 2000.

Abstract: Fault tolerance is a crucial property for recent distributed systems. We propose an algorithm that solves the census problem (list all processor identifiers and their relative distance) on an arbitrary strongly connected network. This algorithm tolerates transient faults that corrupt the processors and communication links memory (it is self-stabilizing) as well as intermittent faults (fair loss, reorder, finite duplication of messages) on communication media. A formal proof establishes its correctness for the considered problem. Our algorithm leads to the construction of algorithms for any silent problems that are self-stabilizing while supporting the same communication hazards.

[DT00cb] Bertrand Ducourthial and Sébastien Tixeuil. Self-stabilization with path algebra. In *International Colloquium on Structural Information and Communication Complexity(Sirroco'2000)*, pages 95–110, L'Aquila, Italy, June 2000. Carleton University Press.

Abstract: Self-stabilizing protocols can resist transient failures and guarantee system recovery in a finite time. We highlight the connexions between the formalism of self-stabilizing distributed systems and the formalism of generalized path algebra and asynchronous iterations with delay. We use the later to prove that a local condition on locally executed algorithm (being a strictly idempotent r-operator) ensures self-stabilization of the global system. As a result, a parametrized distributed algorithm applicable to any directed graph topology is proposed. Its function parameter can be instantiated to produce a large class of protocols, which are self-stabilizing at no additional cost.

[DT00cc] Bertrand Ducourthial and Sébastien Tixeuil. Self-stabilizing construction of optimal broadcast forests. In *Rencontres Francophones sur les aspects Algorithmiques des Télécommunications (AlgoTel'2000)*, pages 149–154, La Rochelle, France, May 2000. in French.

Abstract: Dans cet article, nous nous intéressons à la détermination des arborescences permettant les meilleures diffusions, dans le cas où le critère d'évaluation peut être exprimé localement au niveau de chaque lien, sous la forme d'un taux. Nous proposons un algorithme de construction d'arborescence auto-stabilisant, c'est-à-dire apte à resister aux défaillances temporaires des sites ou des liens de communication.

[DT00r] Sylvie Delaët and Sébastien Tixeuil. Tolerating transient and intermittent failures. Technical Report 1246, University of Paris Sud XI, Laboratoire de Recherche en Informatique, March 2000.

Abstract: Fault tolerance is a crucial property for recent distributed systems. We propose an algorithm that solves the census problem (list all processor identifiers and their relative distance) on an arbitrary strongly connected network. This algorithm tolerates transient faults that corrupt the processors and communication links memory (it is self-stabilizing) as well as intermittent faults (fair loss, reorder, finite duplication of messages) on communication media. A formal proof establishes its correctness for the considered problem. Our algorithm leads to the construction of algorithms for any silent problems that are self-stabilizing while supporting the same communication hazards.

[DT01c] Bertrand Ducourthial and Sébastien Tixeuil. Adaptive multi-sourced multicast. In *Rencontres Francophones sur les aspects Algorithmiques des Télécommunications (AlgoTel'2001)*, pages 135–142, St-Jean de Luz, France, May 2001. in French.

Abstract: Dans cet article, nous nous intéressons à la détermination d'un schéma de multidistribution permettant les meilleures retransmissions, dans le cas où le critère d'évaluation peut être exprimé localement au niveau de chaque lien, sous la forme d'une métrique habituelle (poids, taux, etc.). Nous proposons un algorithme de construction d'une forêt de multidistribution qui est auto-stabilisant, c'est-à-dire qui résiste aux défaillances temporaires des sites ou des liens de communication, et qui tolère des aléas de communication (pertes, duplications, déséquencements). La propriété d'auto-stabilisation lui permet, sous certaines conditions, de s'adapter à une nouvelle configuration (topologie, variation des poids ou des taux, modification des groupes) sans avoir à écrire de code spécifique.

[DT01ja] Ajoy K Datta and Sébastien Tixeuil. Self-stabilizing sorting on tree networks. *Parallel Algorithms and Applications*, 16(1):1–15, January 2001.

Abstract: This paper presents a self-stabilizing distributed sorting algorithm for tree networks. The distributed sorting problem can be informally described as follows: Nodes cooperate to reach a global configuration where every node, depending on its identifier, is assigned a specific final value taken from a set of input values distributed across all nodes. The input values may change in time. In our solution, the system reaches its final configuration in a finite time after the input values are stable and the faults cease. The fault-tolerance and the adaptivity to changing input is achieved using Dijkstra's paradigm of self-stabilization. A self-stabilizing algorithm,

regardless of the initial system state, will converge in finite time to a set of *legitimate* states without the need for explicit exception handlers or backward recovery. Our solution is based on a continuous broadcast with acknowledgment along the tree edges to achieve the synchronization among processes in the system. It has $O(n \times h)$ time complexity and only $O(log(n) \times \delta)$ memory requirement where δ is the degree of the tree and h is the height of the tree.

[DT01jb] Bertrand Ducourthial and Sébastien Tixeuil. Self-stabilization with r-operators. *Distributed Computing*, 14(3):147–162, July 2001.

Abstract: This paper describes a parametrized distributed algorithm applicable to any directed graph topology. The function parameter of our algorithm is instantiated to produce distributed algorithms for both fundamental and high level applications, such as shortest path calculus and depth-first-search tree construction. Due to fault resilience properties of our algorithm, the resulting protocols are self-stabilizing at no additional cost. Self-stabilizing protocols can resist transient failures and guarantee system recovery in a finite time. Since the condition on the function parameter (being a strictly idempotent r-operator) permits a broad range of applications to be implemented, the solution presented in our paper can be useful for a large class of distributed systems.

[DT01r] Bertrand Ducourthial and Sébastien Tixeuil. Self-stabilization with r-operators. Technical Report 2001/358, Université de Technologie de Compiègne, HEUDIASYC, 2001.

Abstract: This paper describes a parametrized distributed algorithm applicable to any directed graph topology. The function parameter of our algorithm is instantiated to produce distributed algorithms for both fundamental and high level applications, such as shortest path calculus and depth-first-search tree construction. Due to fault resilience properties of our algorithm, the resulting protocols are self-stabilizing at no additional cost. Self-stabilizing protocols can resist transient failures and guarantee system recovery in a finite time. Since the condition on the function parameter (being a strictly idempotent r-operator) permits a broad range of applications to be implemented, the solution presented in our paper can be useful for a large class of distributed systems.

[DT02j] Sylvie Delaët and Sébastien Tixeuil. Tolerating transient and intermittent failures. *Journal of Parallel and Distributed Computing*, 62(5):961–981, May 2002.

Abstract: Fault tolerance is a crucial property for recent distributed systems. We propose an algorithm that solves the census problem (list all processor identifiers and their relative distance) on an arbitrary strongly connected network. This algorithm tolerates transient faults that corrupt the processors and communication links memory (it is self-stabilizing) as well as intermittent faults (fair loss, reorder, finite duplication of messages) on communication media. A formal proof establishes its correctness for the considered problem. Our algorithm leads to the construction of algorithms for any silent problems that are self-stabilizing while supporting the same communication hazards.

[DT03j] Bertrand Ducourthial and Sébastien Tixeuil. Self-stabilization with path algebra. *Theoretical Computer Science*, 293(1):219–236, 2003. Extended abstract in Sirrocco 2000.

Abstract: Self-stabilizing protocols can resist transient failures and guarantee system recovery in a finite time. We highlight the connexion between the formalism of self-stabilizing distributed systems and the formalism of generalised path algebra and asynchronous iterations with delay. We use the later to prove that a local condition on locally executed algorithm (being a strictly idempotent r-operator) ensures self-stabilization of the global system. As a result, a parametrized distributed algorithm applicable to any directed graph topology is proposed, and the function parameter of our algorithm is instantiated to produce distributed algorithms for both fundamental and high level applications. Due to fault resilience properties of our algorithm, the resulting protocols are self-stabilizing at no additional cost.

[DT97c] Sylvie Delaët and Sébastien Tixeuil. Auto-stabilisation en dépit de communications non fiables. In *Renpar'9*, Lausanne, Swizerland, June 1997.

Abstract: La tolérance aux défaillances est une propriété cruciale pour les systèmes répartis actuels. Nous proposons un algorithme résolvant le problème de recensement (répertorier les processeurs présents et les localiser) sur un anneau unidirectionnel. Cet algorithme tolère les défaillances transitoires sur les mémoires des processeurs et le contenu des liens de communication (il est auto-stabilisant), ainsi que les défaillances intermittentes (perte équitable, déséquencement, duplication finie des messages) sur les liens de communication. Une preuve formelle l'accompagne établissant sa correction pour le problème considéré. En plus de confirmer l'existence d'algorithmes auto-stabilisants en dépit de liens

non fiables, notre algorithme induit la construction de nouveaux algorithmes auto-stabilisants résolvant plusieurs problèmes fondamentaux et supportant les mêmes aléas de communication.

[DT98c] Bertrand Ducourthial and Sébastien Tixeuil. Selfstabilizing global computations with r-operators. In Alain Bui and Vincent Villain, editors, *Distributed Computing, 2nd International Conference On Principles Of Distributed Systems. OPODIS 98, Amiens, France, December 16-18, 1998*, pages 99–114. Hermes, 1998.

Abstract: This paper describes a parametrized distributed algorithm applicable to any directed graph topology. The function parameter of our algorithm is instantiated to produce distributed algorithms for both fundamental and high level applications, such as shortest path calculus and depth-first-search tree construction. Due to fault resilience properties of our algorithm, the resulting protocols are self-stabilizing at no additional cost. Self-stabilizing protocols can resist transient failures and guarantee system recovery in a finite time. Since the condition on the function parameter (being a strictly idempotent r-operator) permits a broad range of applications to be implemented, we believe the solution presented in our paper can be quite useful for a large class of distributed systems.

[DT98j] Sylvie Delaët and Sébastien Tixeuil. Un algorithme auto-stabilisant en dépit de communications non fiables. *Technique et Science Informatiques*, 5(17), 1998.

Abstract: Fault tolerance is a crucial property for recent distributed systems. We propose an algorithm that solves the census problem (list all processor identifiers and locate them) on a unidirectionnal ring. This algorithm tolerates transient faults that corrupt the processors and communication links memory (it is self-stabilizing) as well as intermittent faults (fair loss, desequencing, finite duplication of messages) on communication media. This algorithm is followed by a formal proof establishing its correctness for the considered problem. In addition to confirm the existence of self-stabilizing algorithm in spite of unreliable communication media, our algorithm leads to the building of new self-stabilizing algorithms for other fundamental problems while supporting the same communication hazards.

[DT98r] Bertrand Ducourthial and Sébastien Tixeuil. Self-stabilizing global computations with r-operators. Technical Report 1182, Laboratoire de Recherche en Informatique, July 1998.

Abstract: This paper describes a parametrized distributed algorithm applicable to any directed graph topology. The function pa-

rameter of our algorithm is instantiated to produce distributed algorithms for both fundamental and high level applications, such as shortest path calculus and depth-first-search tree construction. Due to fault resilience properties of our algorithm, the resulting protocols are self-stabilizing at no additional cost. Self-stabilizing protocols can resist transient failures and guarantee system recovery in a finite time. Since the condition on the function parameter (being a strictly idempotent r-operator) permits a broad range of applications to be implemented, we believe the solution presented in our paper can be quite useful for a large class of distributed systems.

[DT99r] Bertrand Ducourthial and Sébastien Tixeuil. Self-stabilization with path algebra. Technical Report 1202, University of Paris Sud XI, Laboratoire de Recherche en Informatique, February 1999.

Abstract: Self-stabilizing protocols can resist transient failures and guarantee system recovery in a finite time. We highlight the connexions between the formalism of self-stabilizing distributed systems and the formalism of generalized path algebra and asynchronous iterations with delay. We use the later to prove that a local condition on locally executed algorithm (being a strictly idempotent r-operator) ensures self-stabilization of the global system. As a result, a parametrized distributed algorithm applicable to any directed graph topology is proposed. Its function parameter can be instantiated to produce a large class of protocols, which are self-stabilizing at no additional cost.

[FIRT05c] Pierre Fraigniaud, David Ilcinkas, Sergio Rajsbaum, and Sébastien Tixeuil. Space lower bounds for graph exploration via reduced automata. In Andrzej Pelc and Michel Raynal, editors, *Structural Information and Communication Complexity, 12th International Colloquium, SIROCCO 2005, Mont Saint-Michel, France, May 24-26, 2005, Proceedings*, volume 3499 of *Lecture Notes in Computer Science*, pages 140–154. Springer Verlag, May 2005.

Abstract: We consider the task of exploring graphs with anonymous nodes by a team of non-cooperative robots modeled as finite automata. These robots have no *a priori* knowledge of the topology of the graph, or of its size. Each edge has to be traversed by at least one robot. We first show that, for any set of q non-cooperative K-state robots, there exists a graph of size $O(qK)$ that no robot of this set can explore. This improves the $O(K^{O(q)})$ bound by Rollik (1980). Our main result is an application of this improvement. It concerns exploration with stop, in which one robot has to explore and stop after completing exploration. For this task, the robot is provided with a pebble, that it can use to mark nodes. We prove that exploration with stop requires $\Omega(\log n)$ bits for the family of graphs with at most n nodes. On the other hand, we prove that there exists an exploration with stop algorithm using a robot with $O(D \log \Delta)$ bits of memory to explore all graphs of diameter at most D and degree at most Δ.

[FIRT06b] Pierre Fraigniaud, David Ilcinkas, Sergio Rajsbaum, and Sébastien Tixeuil. *Shimon Even Festschrift*, chapter The reduced automata technique for graph exploration space lower bounds, pages 1–26. Number 3895 in Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg, 2006.

Abstract: We consider the task of exploring graphs with anonymous nodes by a team of non-cooperative robots, modeled as finite automata. For exploration to be completed, each edge of the graph has to be traversed by at least one robot. In this paper, the robots have no a priori knowledge of the topology of the graph, nor of its size, and we are interested in the amount of memory the robots need to accomplish exploration, We introduce the so-called reduced automata technique, and we show how to use this technique for deriving several space lower bounds for exploration. Informally speaking, the reduced automata technique consists in reducing a robot to a simpler form that preserves its "core" behavior on some graphs. Using this technique, we first show that any set of $q \geq 1$ non-cooperative robots, requires $\Omega(\log(n/q))$ memory bits to explore all n-node graphs. The proof implies that, for any set of q K-state robots, there exists a graph of size $O(qK)$ that no robot of this set can explore, which improves the $O(K^{O(q)})$ bound by Rollik (1980). Our main result is an application of this latter result, concerning terminating graph exploration with one robot, i.e., in which the robot is requested to stop after completing exploration. For this task, the robot is provided with a pebble, that it can use to mark nodes (without such a marker, even terminating exploration of cycles cannot be achieved). We prove that terminating exploration requires $\Omega(\log n)$ bits of memory for a robot achieving this task in all n-node graphs.

[GT00c] Maria Gradinariu and Sébastien Tixeuil. Self-stabilizing vertex coloring of arbitrary graphs. In *International Conference on Principles of Distributed Systems (OPODIS'2000)*, pages 55–70, Paris, France, December 2000.

Abstract: A self-stabilizing algorithm, regardless of the initial system state, converges in finite time to a set of states that satisfy a legitimacy predicate without the need for explicit exception handler of backward recovery. The vertex coloration problem consists

in ensuring that every node in the system has a color that is different from any of its neighbors. We provide three self-stabilizing solutions to the vertex coloration problem under unfair scheduling that are based on a greedy technique. We use at most $B+1$ different colors (in complete graphs), where B is the node degree, and ensure stabilization within $O(n \times B)$ processor atomic steps. Two of our algorithms deal with uniform networks where nodes do not have identifiers. Our solutions lead to directed acyclic orientation and maximal independent set construction at no additional cost.

[GT00ra] Maria Gradinariu and Sébastien Tixeuil. Self-stabilizing vertex coloring of arbitrary graphs. Technical Report 1260, Laboratoire de Recherche en Informatique, University of Paris Sud XI, September 2000.

Abstract: A self-stabilizing algorithm, regardless of the initial system state, converges in finite time to a set of states that satisfy a legitimacy predicate without the need for explicit exception handler of backward recovery. The vertex coloration problem consists in ensuring that every node in the system has a color that is different from any of its neighbors. We provide three self-stabilizing solutions to the vertex coloration problem under unfair scheduling that are based on a greedy technique. We use at most $B+1$ different colors (in complete graphs), where B is the graph degree, and ensure stabilization within $O(n \times B)$ processor atomic steps. Two of our algorithms deal with uniform networks where nodes do not have identifiers. Our solutions lead to directed acyclic orientation and maximal independent set construction at no additional cost.

[GT00rb] Maria Gradinariu and Sébastien Tixeuil. Tight space uniform self-stabilizing l-mutual exclusion. Technical Report 1249, University of Paris Sud XI, Laboratoire de Recherche en Informatique, March 2000.

Abstract: A self-stabilizing algorithm, regardless of the initial system state, converges in finite time to a set of states that satisfy a legitimacy predicate without the need for explicit exception handler of backward recovery. The l-mutual exclusion is a generalization of the fundamental problem of mutual exclusion : the system has to guarantee the fair sharing of a resource that can be used by l processors simultaneously. We present a space efficient solution to the l-mutual exclusion problem that performs on uniform unidirectional ring networks and that is self-stabilizing. Our solution improves the space complexity of previously known approaches by a factor of $\min(n^2 \times \log(n), \frac{1}{l} \times \log^{l-1}(n))$, while retaining none of their drawbacks in terms of system hypothesis (we support unfair

scheduler and ensure strong correctness) or specification verification (we guarantee high level l-mutual exclusion). When l is fixed, the space complexity at each node is constant in average, making our approach suitable for scalable systems.

[GT01c] Maria Gradinariu and Sébastien Tixeuil. Tight space uniform self-stabilizing l-mutual exclusion. In *IEEE International Conference on Distributed Computing Systems (ICDCS'01)*, pages 83–90, Phoenix, Arizona, May 2001. IEEE Press.

Abstract: A self-stabilizing algorithm, regardless of the initial system state, converges in finite time to a set of states that satisfy a legitimacy predicate without the need for explicit exception handler of backward recovery. The l-mutual exclusion is a generalization of the fundamental problem of mutual exclusion : the system has to guarantee the fair sharing of a resource that can be used by l processors simultaneously. We present a space efficient solution to the l-mutual exclusion problem that performs on uniform unidirectional ring networks and that is self-stabilizing. Our solution improves the space complexity of previously known approaches by a factor of $\min(n^2 \times \log(n), \frac{1}{l} \times \log^{l-1}(n))$, while retaining none of their drawbacks in terms of system hypothesis (we support unfair scheduler and ensure strong correctness) or specification verification (we guarantee high level l-mutual exclusion). When l is fixed, the space complexity at each node is constant in average, making our approach suitable for scalable systems.

[GT01r] Christophe Genolini and Sébastien Tixeuil. Reactive k-stabilization and time adaptivity: possibility and impossibility results. Technical Report 1276, Laboratoire de Recherche en Informatique, University of Paris Sud XI, 2001.

Abstract: It is desirable that the smaller is the number of faults to hit a network, the faster should a network protocol recover. We study the scenario where up to k (for a given k) faults hit processors of a synchronous distributed system by corrupting their state undetectably. The exact number of faults, the specific faulty nodes, and the time the faults hit are *not* known. In this context, we first present a motivation for studying time adaptive protocols (*i.e.* protocols that recover from memory corruption in a time that depends only on the number of faults and *not* on the network size) in the synchronous model. In more details, we prove that when using the usual asynchronous step model, there exists no time adaptive protocol for any non-trivial dynamic problem, even if we assume that the number of faults is bounded by 1. Then, we present a solution for the reactive

problem that is often used as a benchmark in this setting: the problem of *token passing*. Unlike previous approaches, we consider uniform ring networks where processors execute deterministic code. It is known that such a setting makes self-stabilizing solutions (*i.e.* that tolerate an arbitrary number of faults) impossible. We present a solution for the same setting, provided that no more than k faults hit the system (and k is lower than , where n is the size of the system). Our algorithm has the additional property of being time adaptive: the time needed to recover from failures is proportional to the actual number of faults.

[GT02c] Christophe Genolini and Sébastien Tixeuil. A lower bound on k-stabilization in asynchronous systems. In *Proceedings of IEEE 21st Symposium on Reliable Distributed Systems (SRDS'2002)*, Osaka, Japan, October 2002.

Abstract: It is desirable that the smaller is the number of faults to hit a network, the faster should a network protocol recover. We study the scenario where up to k (for a given k) faults hit processors of a synchronous distributed system by corrupting their state undetectably. In this context, we show that the well known step complexity model is not appropriate to study time complexity of time-adaptive protocols (*i.e.* protocols that recover from memory corruption in a time that depends only on the number of faults and *not* on the network size). In more details, we prove that for non trivial dynamic problems (such as token passing), there exists a lower bound of $\Omega(D)$ (where D is the network diameter) steps on the stabilization time even when as few as 1 corruption hits the system. This implies that there exist no time adaptive protocol for those problems in the asynchronous step model, even if we assume that the number of faults is bounded by 1 and that the scheduling of the processors is almost synchronous (between two actions of an enabled processor, any other processor may execute at most D actions).

[GT06r] Maria Gradinariu and Sébastien Tixeuil. Conflict managers for self-stabilization without fairness assumption. Technical Report 1459, LRI, Université Paris Sud, September 2006.

URL `http://www.lri.fr/Rapports-internes/2006/RR1459.pdf`

Abstract: In this paper, we specify the *conflict manager* abstraction. Informally, a conflict manager guarantees that any two neighboring nodes can not enter their critical simultaneously (*safety*), and that at least one node is able to execute its critical section (*progress*). The conflict manager problem is strictly weaker than the classical local mutual exclusion problem, where any node that requests to enter its critical section eventually does so (*fairness*). We argue that conflict managers are a useful mechanism to transform a large class of self-stabilizing algorithms that operate in an essentially sequential model, into self-stabilizing algorithm that operate in a completely asynchronous distributed model. We provide two implementations (one deterministic and one probabilistic) of our abstraction, and provide a composition mechanism to obtain a generic transformer. Our transformers have low overhead: the deterministic transformer requires one memory bit, and guarantees time overhead in order of the network degree, the probabilistic transformer does not require extra memory. While the probabilistic algorithm performs in anonymous networks, it only provides probabilistic stabilization guarantees. In contrast, the deterministic transformer requires initial symmetry breaking but preserves the original algorithm guarantees.

[GT07ca] Fabíola Greve and Sébastien Tixeuil. Knowledge connectivity vs. synchrony requirements for fault-tolerant agreement in unknown networks. In *Proceedings of IEEE International Conference on Dependable Systems and networks (DSN 2007)*, page to appear. IEEE, June 2007.

URL `https://hal.inria.fr/inria-00123020`

Abstract: In self-organizing systems, such as mobile ad-hoc and peer-to-peer networks, consensus is a fundamental building block to solve agreement problems. It contributes to coordinate actions of nodes distributed in an *ad-hoc* manner in order to take consistent decisions. It is well known that in classical environments, in which entities behave asynchronously and where identities are known, consensus cannot be solved in the presence of even one process crash. It appears that self-organizing systems are even less favorable because the set and identity of participants are not known. We define necessary and sufficient conditions under which fault-tolerant consensus become solvable in these environments. Those conditions are related to the synchrony requirements of the environment, as well as the connectivity of the knowledge graph constructed by the nodes in order to communicate with their peers.

[GT07cb] Maria Gradinariu and Sébastien Tixeuil. Conflict managers for self-stabilization without fairness assumption. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS 2007)*. IEEE, June 2007.

URL `http://www.lri.fr/Rapports-internes/2006/RR1459.pdf`

Abstract: In this paper, we specify the *conflict manager* abstraction. Informally, a conflict manager guarantees that any two neighboring nodes can not enter their critical simultaneously (*safety*), and that at least one node is able to execute its critical section (*progress*). The conflict manager problem is strictly weaker than the classical local mutual exclusion problem, where any node that requests to enter its critical section eventually does so (*fairness*). We argue that conflict managers are a useful mechanism to transform a large class of self-stabilizing algorithms that operate in an essentially sequential model, into self-stabilizing algorithm that operate in a completely asynchronous distributed model. We provide two implementations (one deterministic and one probabilistic) of our abstraction, and provide a composition mechanism to obtain a generic transformer. Our transformers have low overhead: the deterministic transformer requires one memory bit, and guarantees time overhead in order of the network degree, the probabilistic transformer does not require extra memory. While the probabilistic algorithm performs in anonymous networks, it only provides probabilistic stabilization guarantees. In contrast, the deterministic transformer requires initial symmetry breaking but preserves the original algorithm guarantees.

[GT07cc] Fabíola Gonçalves Pereira Greve and Sébastien Tixeuil. Condições de conectividade para realização de acordo tolerante a falhas em sistemas auto-organizaveis. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2007)*, May 2007.

Abstract: O consenso e um problema fundamental para o desenvolvimento de soluções confiaveis em sistemas auto-organizaveis, tais como redes moveis adhoc e de sensores. Ele permite com que nos distribuıdos de maneira ad-hoc possam sincronizar as suas ações com o intuito de alcançar decisões comuns. Infelizmente, em ambientes classicos assíncronos, onde o conjunto dos participantes e conhecido, o problema do consenso não tem solução determinística na presença de uma única falha de processo. Assim, espera-se que a resolução deste problema em sistemas onde os participantes são desconhecidos, tenha uma complexidade ainda maior. Neste artigo, estudamos condições necessárias e suficientes e propomos algoritmos para a resolução do consenso tolerante a falhas em sistemas auto-organizáveis. Tais condições são relacionadas, por um lado, aos requisitos de conectividade do grafo de conhecimento construído pelos nos para comunicação com os seus pares, e por outro lado, aos requisitos de sincronia a serem incorporados ao sistema.

[GT07r] Fabiola Greve and Sébastien Tixeuil. Knowledge connectivity vs. synchrony requirements for fault-tolerant agreement in unknown networks. Research Report 6099, INRIA, January 2007.

URL `https://hal.inria.fr/inria-00123020`

Abstract: In self-organizing systems, such as mobile ad-hoc and peer-to-peer networks, consensus is a fundamental building block to solve agreement problems. It contributes to coordinate actions of nodes distributed in an *ad-hoc* manner in order to take consistent decisions. It is well known that in classical environments, in which entities behave asynchronously and where identities are known, consensus cannot be solved in the presence of even one process crash. It appears that self-organizing systems are even less favorable because the set and identity of participants are not known. We define necessary and sufficient conditions under which fault-tolerant consensus become solvable in these environments. Those conditions are related to the synchrony requirements of the environment, as well as the connectivity of the knowledge graph constructed by the nodes in order to communicate with their peers.

[HHLRT06r] Thomas Herault, William Hoarau, Pierre Lemarinier, Eric Rodriguez, and Sébastien Tixeuil. Fail-mpi: How fault-tolerant is fault-tolerant mpi? Technical Report 1450, Université Paris-Sud XI, June 2006.

Abstract: One of the topics of paramount importance in the development of Cluster and Grid middleware is the impact of faults since their occurrence probability in a Grid infrastructure and in large-scale distributed system is actually very high. MPI (Message Passing Interface) is a popular abstraction for programming distributed computation applications. FAIL is an abstract language for fault occurrence description capable of expressing complex and realistic fault scenarios. In this paper, we investigate the possibility of using FAIL to inject faults in a fault-tolerant MPI implementation. Our middleware, FAIL-MPI, is used to carry quantitative and qualitative faults and stress testing.

[HLHRTC06c] William Hoarau, Pierre Lemarinier, Thomas Herault, Eric Rodriguez, Sébastien Tixeuil, and Franck Cappello. Fail-mpi: How fault-tolerant is fault-tolerant mpi? In *Proceedings of Cluster 2006*, Barcelona, Spain, September 2006.

Abstract: One of the topics of paramount importance in the development of Cluster and Grid middleware is the impact of faults since their occurrence probability in a Grid infrastructure and in large-scale distributed system is actually very high. MPI (Message Passing Interface) is a popular abstraction for programming distributed computation applications. FAIL is an abstract language for fault occurrence description capable of expressing complex and realistic fault scenarios. In this paper, we investigate the possibility of using FAIL to inject faults in a fault-tolerant MPI implementation. Our middleware, FAIL-MPI, is used to carry quantitative and qualitative faults and stress testing.

[HST06bc] William Hoarau, Luis Silva, and Sébastien Tixeuil. *Integrated Research in Grid Computing*, chapter Fault-injection and Dependability Benchmarking for GRID COmputing Middleware. CoreGRID. Springer Verlag, 2006.

Abstract: We present the state-of-the-art about fault-injection and dependability benchmarking and we explain the importance of this kind of tools for dependability assessment of Grid-based applications and Grid middleware. Our emphasis goes to the FAIL-FCI fault-injection software that has been de-veloped in INRIA Grand Large, and a benchmark tool called QUAKE that was developed by the University of Coimbra. We present some experimental results taken with these two tools.

[HST06r] William Hoarau, Luis Silva, and Sébastien Tixeuil. An overview of existing tools for fault-injection and dependability benchmarking in grids. Technical Report 0041, CoreGRID, October 2006.

URL `http://www.coregrid.net/mambo/images/stories/TechnicalReport`

Abstract: In this paper we review several existing tools for fault injection and dependability benchmarking in grids. We emphasis on the FAIL-FCI fault-injection software that has been developed in INRIA Grand Large, and a benchmark tool called QUAKE that has been developed in the University of Coimbra. We present the state-of-the-art and we explain the importance of these tools for dependability assessment of Grid-based applications and Grid middleware.

[HT03r] Ted Herman and Sébastien Tixeuil. A distributed tdma slot assignment algorithm for wireless sensor networks. Technical Report 1370, Laboratoire de Recherche en Informatique, 2003.

URL `http://arxiv.org/abs/cs.DC/0405042`

Abstract: Wireless sensor networks benefit from communication protocols that reduce power requirements by avoiding frame collision. Time Division Media Access methods schedule transmission in slots to avoid collision, however these methods often lack scalability when implemented in ad hoc networks subject to node failures and dynamic topology. This paper reports a distributed algorithm for TDMA slot assignment that is self-stabilizing to transient faults and dynamic topology change. The expected convergence time is $O(1)$ for any size network satisfying a constant bound on the size of a node neighborhood.

[HT04c] Ted Herman and Sébastien Tixeuil. A distributed tdma slot assignment algorithm for wireless sensor networks. In *Proceedings of the First Workshop on Algorithmic Aspects of Wireless Sensor Networks (AlgoSensors'2004)*, number 3121 in Lecture Notes in Computer Science, pages 45–58, Turku, Finland, July 2004. Springer-Verlag.

Abstract: Wireless sensor networks benefit from communication protocols that reduce power requirements by avoiding frame collision. Time Division Media Access methods schedule transmission in slots to avoid collision, however these methods often lack scalability when implemented in *ad hoc* networks subject to node failures and dynamic topology. This paper reports a distributed algorithm for TDMA slot assignment that is self-stabilizing to transient faults and dynamic topology change. The expected local convergence time is $O(1)$ for any size network satisfying a constant bound on the size of a node neighborhood.

[HT04cb] Ted Herman and Sébastien Tixeuil. Un algorithme tdma réparti pour les réseaux de capteurs. In *Proceedings of Rencontres Francophones sur l'Algorithmique des Communications (Algotel'2004)*, Batz sur Mer, France, 2004. ENST Bretagne.

Abstract: Les méthodes TDMA utilisent le multiplexage temporel et planifient les transmissions pour éviter les collisions de trames. Cependant, ces méthodes souffrent d'un problème de dimensionnement quand elles sont utilisées dans des réseaux *ad hoc* à topologie dynamique et sujets à des défaillances de nœuds. Cet article propose un algorithme réparti pour la planification TDMA qui résiste de manière auto-stabilisante aux défaillances transitoires et aux changements dynamiques de topologie. Le temps de convergence est $O(1)$ indépendamment de la taille du réseau si celui-ci admet une borne constante sur la taille du voisinage de chaque nœud.

[HT05c] William Hoarau and Sébastien Tixeuil. A language-driven tool for fault injection in distributed applications. In *Proceedings of the IEEE/ACM Workshop GRID 2005*, page to appear, Seattle, USA, November 2005.

Abstract: In a network consisting of several thousands computers, the occurrence of faults is unavoidable. Being able to test the behavior of a distributed program in an environment where we can control the faults (such as the crash of a process) is an important feature that matters in the deployment of reliable programs. In this paper, we present FAIL (for FAult Injection Language), a language that permits to elaborate complex fault scenarios in a simple way, while relieving the user from writing low level code. Besides, it is possible to construct probabilistic scenarios (for average quantitative tests) or deterministic and reproducible scenarios (for studying the application's behavior in particular cases). We also present FCI, the FAIL Cluster Implementation, that consists of a compiler, a runtime library and a middleware platform for software fault injection in distributed applications. FCI is able to interface with numerous programming languages without requiring the modification of their source code, and the preliminary tests that we conducted show that its effective impact at runtime is low.

[HT05p] Ted Herman and Sébastien Tixeuil, editors. *Self-stabilizing Systems*, volume 3764 of *Lecture Notes in Computer Science*, Barcelona, Spain, October 2005. Springer Verlag.

URL `http://www.springeronline.com/3-540-29814-2`

Abstract: This book constitutes the refereed proceedings of the 7th International Symposium on Self-Stabilizing Systems, SSS 2005, held in Barcelona, Spain, in October 2005. The 15 revised full papers presented were carefully reviewed and selected from 33 submissions. The papers address classical topics of self-stabilization, prevailing extensions to the field, such as snap-stabilization, code stabilization, self-stabilization with either dynamic, faulty or Byzantine components, or deal with applications of self-stabilization, either related to operating systems, security, or mobile and ad hoc networks.

[HT05ra] William Hoarau and Sébastien Tixeuil. A language-driven tool for fault injection in distributed systems. Technical Report 1399, Laboratoire de Recherche en Informatique, February 2005.

Abstract: In a network consisting of several thousands computers, the occurrence of faults is unavoidable. Being able to test the behavior of a distributed program in an environment where we can control the faults (such as the crash of a process) is an important feature that matters in the deployment of reliable programs. In this paper, we present FAIL (for FAult Injection Language), a language that permits to elaborate complex fault scenarios in a simple way, while relieving the user from writing low level code. Besides, it is possible to construct probabilistic scenarios (for average quantitative tests) or deterministic and reproducible scenarios (for studying the application's behavior in particular cases). We also present FCI, the FAIL Cluster Implementation, that consists of a compiler, a runtime library and a middleware platform for software fault injection in distributed applications. FCI is able to interface with numerous programming languages without requiring the modification of their source code, and the preliminary tests that we conducted show that its effective impact at runtime is low.

[HT06c] William Hoarau and Sébastien Tixeuil. Easy fault injection and stress testing with fail-fci. In *Second CoreGRID Workshop on Grid and Peer to Peer Systems Architecture*, Paris, France, January 2006.

Abstract: In a network consisting of several thousands computers, the occurrence of faults is unavoidable. Being able to test the behavior of a distributed program in an environment where we can control the faults (such as the crash of a process) is an important feature that matters in the deployment of reliable programs. In this paper, we extend FAIL-FCI (for Fault Injection Language, and FAIL Cluster Implementation, respectively), a software tool that permits to elaborate complex fault scenarios in a simple way, while relieving the user from writing low level code. In particular, we show that not only we are able to fault-load existing distributed applications (as used in most current papers that address fault-tolerance issues), we are also able to inject qualitative faults, i.e. inject specific faults at very specific moments in the program code of the application under test. Finally, and although this was not the primary purpose of the tool, we are also able to inject specific patterns of workload, in order to stress test the application under test. Interestingly enough, the whole process is driven by a simple unified description language, that is totally independent from the language of the application, so that no code changes or recompilation are needed on the application side.

[HTMSS06c] William Hoarau, Sébastien Tixeuil, Nuno Moreno, Décio Sousa, and Luis Silva. Benchmarking the ogsa-dai middleware. In *Second Coregrid Integration Workshop*, Krakow, Poland, October 2006.

Abstract: one important contribution to the community that is developing Grid middleware is the definition and implementation of benchmarks and tools to assess the performance and dependability of Grid applications and the corresponding middleware. In this paper, we present an experimental study that was conducted with OGSA-DAI, a popular package of middleware that provides access to remote data resources thought a unified web-service front-end. The results show that OGSA-DAI is quite stable and performed quite well in scalability tests, executed on Grid5000. However, we also demonstrate that OGSA-DAI is currently using a SOAP container (Apache Axis1.3) that suffers from severe memory leaks. We show how the default configuration of OGSA-DAI is not affected by that problem, but a small change in the configuration of a web-service may lead to very unreliable execution of OGSA-DAI.

[HTRSS06r] William Hoarau, Sébastien Tixeuil, Nuno Rodrigues, Décio Sousa, and Luis Silva. Benchmarking the ogsa-dai middleware. Technical Report 0060, CoreGRID, October 2006.

URL `http://www.coregrid.net/mambo/images/stories/TechnicalReport`

Abstract: One important contribution to the community that is developing Grid middleware is the definition and implementation of benchmarks and tools to assess the performance and dependability of Grid applications and the corresponding middleware. In this paper, we present an experimental study that was conducted with OGSA-DAI, a popular package of middleware that provides access to remote data resources thought a unified Web-service front-end. The results show that OGSA-DAI is quite stable and performed quite well in scalability tests, executed on Grid5000. However, we also demonstrate that OGSA-DAI WSI is currently using a SOAP container (Apache Axis1.2.1) that suffers from severe memory leaks. We show how the default configuration of OGSA-DAI is not affected by that problem, but a small change in the configuration of a Web-service may lead to very unreliable execution of OGSA-DAI.

[HTRSS07c] William Hoarau, Sébastien Tixeuil, Nuno Rodrigues, Décio Sousa, and Luis Silva. *Integrated Research in GRID COmputing*, chapter Benchmarking the OGSA-DAI middleware. Springer Verlag, 2007.

URL `http://www.coregrid.net/mambo/images/stories/TechnicalReport`

Abstract: One important contribution to the community that is developing Grid middleware is the definition and implementation of benchmarks and tools to assess the performance and dependability of Grid applications and the corresponding middleware. In this paper, we present an experimental study that was conducted with OGSA-DAI, a popular package of middleware that provides access to remote data resources thought a unified Web-service front-end. The results show that OGSA-DAI is quite stable and performed quite well in scalability tests, executed on Grid5000. However, we also demonstrate that OGSA-DAI WSI is currently using a SOAP container (Apache Axis1.2.1) that suffers from severe memory leaks.

We show how the default configuration of OGSA-DAI is not affected by that problem, but a small change in the configuration of a Web-service may lead to very unreliable execution of OGSA-DAI.

[HTV05ra] William Hoarau, Sébastien Tixeuil, and Fabien Vauchelles. Fault injection in distributed java applications. Technical Report 1420, Laboratoire de Recherche en Informatique, Université Paris Sud, October 2005.

Abstract: In a network consisting of several thousands computers, the occurrence of faults is unavoidable. Being able to test the behaviour of a distributed program in an environment where we can control the faults (such as the crash of a process) is an important feature that matters in the deployment of reliable programs. In this paper, we investigate the possibility of injecting software faults in distributed java applications. Our scheme is by extending the FAIL-FCI software, and does not require any modification of the source code of the application under test, while retaining the possibility to write high level fault scenarios. As a proof of concept, we use our tool to test FreePastry, an existing java implementation of a Distributed Hash Table (DHT), against node failures.

[HTV05rb] William Hoarau, Sébastien Tixeuil, and Fabien Vauchelles. Easy fault injection and stress testing with fail-fci. Technical Report 1421, Laboratoire de Recherche en Informatique, Université Paris Sud, October 2005.

Abstract: In a network consisting of several thousands computers, the occurrence of faults is unavoidable. Being able to test the behavior of a distributed program in an environment where we can control the faults (such as the crash of a process) is an important feature that matters in the deployment of reliable programs. In this paper, we extend FAIL-FCI (for Fault Injection Language, and FAIL Cluster Implementation, respectively), a software tool that permits to elaborate complex fault scenarios in a simple way, while relieving the user from writing low level code. In particular, we show that not only we are able to fault-load existing distributed applications (as used in most current papers that address fault-tolerance issues), we are also able to inject qualitative faults, i.e. inject special faults at very special moments in the program code of the application under test. Finally, and although this was not the primary purpose of the tool, we are also able to inject special patterns of workload, in order to stress test the application under test. Interestingly enough, the whole process is driven by a simple unified description language, that is totally independent from the language of the application, so that no code changes or recompilation are needed on the application side.

[HTV06c] William Hoarau, Sébastien Tixeuil, and Fabien Vauchelles. Fault injection in distributed java applications. In *International Workshop on Java for Parallel and Distributed Computing (joint with IPDPS 2006)*, page to appear, Greece, April 2006. IEEE.

> **Abstract:** In a network consisting of several thousands computers, the occurrence of faults is unavoidable. Being able to test the behaviour of a distributed program in an environment where we can control the faults (such as the crash of a process) is an important feature that matters in the deployment of reliable programs. In this paper, we investigate the possibility of injecting software faults in distributed java applications. Our scheme is by extending the FAIL-FCI software. It does not require any modification of the source code of the application under test, while retaining the possibility to write high level fault scenarios. As a proof of concept, we use our tool to test FreePastry, an existing java implementation of a Distributed Hash Table (DHT), against node failures.

[HTV07j] William Hoarau, Sébastien Tixeuil, and Fabien Vauchelles. Fail-fci: Versatile fault-injection. *Future Generation Computer Systems*, 2007.

> **Abstract:** One of the topics of paramount importance in the development of Grid middleware is the impact of faults since their probability of occurrence in a Grid infrastructure and in large-scale distributed system is actually very high. In this paper we explore the versatility of a new tool for fault injection in distributed applications: FAIL-FCI. In particular, we show that not only we are able to fault-load existing distributed applications (as used in most current papers that address fault-tolerance issues), we are also able to inject *qualitative faults*, *i.e.* inject specific faults at very specific moments in the program code of the application under test. Finally, and although this was not the primary purpose of the tool, we are also able to inject specific patterns of workload, in order to stress test the application under test. Interestingly enough, the whole process is driven by a simple unified description language, that is totally independent from the language of the application, so that no code changes or recompilation are needed on the application side.

[JADT02j] Colette Johnen, Luc Alima, Ajoy K. Datta, and Sébastien Tixeuil. Optimal snap-stabilizing neighborhood synchronizer in tree networks. *Parallel Processing Letters*, 12(3-4):327–340, 2002.

> **Abstract:** We propose a snap-stabilizing synchronization technique, called the *Neighborhood Synchronizer* ($\mathcal{NS}$) that synchronizes nodes

with their neighbors in a tree network. The $\mathcal{N}S$ scheme has optimal memory requirement — only one bit per processor. $\mathcal{N}S$ is *snap-stabilizing*, meaning that it always behaves according to its specification. The proposed synchronizer being snap-stabilizing is optimal in terms of stabilization time. We show an application of the synchronizer by designing an efficient broadcast algorithm ($\mathcal{B}A$) in tree networks. $\mathcal{B}A$ is also snap-stabilizing and needs only $2h + 2m - 1$ rounds to broadcast m messages, where h is the height of the tree.

[JADT98r] Colette Johnen, Luc Alima, Ajoy Kumar Datta, and Sébastien Tixeuil. Self-stabilizing neighborhood synchronizer in tree networks. INFO RR 98-12, Université Catholique de Louvain, November 1998.

Abstract: We propose a self-stabilizing synchronization technique, called the *Neighborhood Synchronizer* ($\mathcal{NS}$), that synchronizes nodes with their neighbors in a tree network. The $\mathcal{NS}$ scheme has extremely small memory requirement—only 1 bit per processor. Algorithm $\mathcal{NS}$ is inherently self-stabilizing. We show an application of the synchronizer to design a broadcasting algorithm $\mathcal{BA}$ in tree networks. Algorithm $\mathcal{BA}$ is also inherently self-stabilizing and needs only $2h+2m-1$ time to broadcast m messages, where h is the height of the tree.

[JADT99c] Colette Johnen, Luc Onana Alima, Ajoy Kumar Datta, and Sébastien Tixeuil. Self-stabilizing neighborhood synchronizer in tree networks. In *Proceedings of the 19th International Conference on Distributed Computing Systems*, pages 487–494, Austin, TX, USA, May 1999. IEEE Computer Society.

Abstract: We propose a self-stabilizing synchronization technique, called the *Neighborhood Synchronizer* ($\mathcal{NS}$), that synchronizes nodes with their neighbors in a tree network. The $\mathcal{NS}$ scheme has extremely small memory requirement—only 1 bit per processor. Algorithm $\mathcal{NS}$ is inherently self-stabilizing. We apply our synchronizer to design a broadcasting algorithm $\mathcal{BA}$ in tree networks. Algorithm $\mathcal{BA}$ is also inherently self-stabilizing and needs only $2h + 2m - 1$ rounds to broadcast m messages, where h is the height of the tree.

[JPT03r] Colette Johnen, Franck Petit, and Sébastien Tixeuil. Auto-stabilisation et protocoles réseau. Technical Report 1357, Laboratoire de Recherche en Informatique, 2003.

Abstract: In 1974, E.W. Dijkstra defined self-stabilization as the property for a distributed system to recover by itself a correct behavior in a finite number of steps, starting from any initial state.

Thus, self-stabilization is a simple and efficient way to tolerate transient faults or failures. This paper surveys works that propose self-stabilizing solutions to problems arising in Computer Networks, such as routing and transport layers, or network control protocols. We also review technics to design self-stabilizing protocols, and self-stabilization refinements that provide fast stabilization time when the number of faults that hit the network is small.

[JPT04j] Colette Johnen, Franck Petit, and Sébastien Tixeuil. Auto-stabilisation et protocoles réseaux. *Technique et Science Informatiques*, 23(8):1027–1056, 2004.

Abstract: In 1974, E.W. Dijkstra defined self-stabilization as the property for a distributed system to recover by itself a correct behavior in a finite number of steps, starting from any initial state. Thus, self-stabilization is a simple and efficient way to tolerate transient faults or failures. This paper surveys works that propose self-stabilizing solutions to problems arising in Computer Networks, such as routing and transport layers, or network control protocols. We also review techniques to design self-stabilizing protocols, and mechanisms that reduce the stabilization time when the number of hitting faults is small.

[JT03c] Colette Johnen and Sébastien Tixeuil. Route preserving stabilization. In *Proceedings of the Sixth Symposium on Self-stabilizing Systems (SSS'03)*, Lecture Notes in Computer Science, San Francisco, USA, June 2003. Springer Verlag. Also in the Proceedings of DSN'03 as a one page abstract.

Abstract: A distributed system is *self-stabilizing* if it returns to a legitimate state in a finite number of steps regardless of the initial state, and the system remains in a legitimate state until another fault occurs. A routing algorithm is *loop-free* if, a path being constructed between two processors p and q, any edges cost change induces a modification of the routing tables in such a way that at any time, there always exists a path from p to q. We present a self-stabilizing loop-free routing algorithm that is also *route preserving*. This last property means that, a tree being constructed, any message sent to the root is received in a bounded amount of time, even in the presence of continuous edge cost changes. Also, and unlike previous approaches, we do not require that a bound on the network diameter is known to the processors that perform the routing algorithm. We guarantee self-stabilization for many metrics (such as minimum distance, shortest path, best transmitter, depth first seach metrics, etc.), by reusing previous results on r-operators.

[JT03r] Colette Johnen and Sébastien Tixeuil. Route preserving stabilization. Technical Report 1353, Laboratoire de Recherche en Informatique, 2003.

Abstract: A distributed system is *self-stabilizing* if it returns to a legitimate state in a finite number of steps regardless of the initial state, and the system remains in a legitimate state until another fault occurs. A routing algorithm is *loop-free* if, a path being constructed between two processors p and q, any edges cost change induces a modification of the routing tables in such a way that at any time, there always exists a path from p to q. We present a self-stabilizing loop-free routing algorithm that is also *route preserving*. This last property means that, a tree being constructed, any message sent to the root is received in a bounded amount of time, even in the presence of continuous edge cost changes. Also, and unlike previous approaches, we do not require that a bound on the network diameter is known to the processors that perform the routing algorithm. We guarantee self-stabilization for many metrics (such as minimum distance, shortest path, best transmitter, depth first seach metrics, etc.), by reusing previous results on r-operators.

[JTDA98r] Colette Johnen, Sébastien Tixeuil, Ajoy Kumar Datta, and Luc Alima. Self-stabilization with causally synchronized wave on tree networks. Technical Report 1144, Laboratoire de Recherche en Informatique, January 1998.

Abstract: We propose a self-stabilizing synchronization technique, called the *Neighborhood Synchronizer* ($\mathcal{NS}$), that synchronizes nodes with their neighbors in a tree network. The $\mathcal{NS}$ scheme has extremely small memory requirement—only 1 bit per processor. Algorithm $\mathcal{NS}$ is inherently self-stabilizing. We show an application of the synchronizer to design a broadcasting algorithm $\mathcal{BA}$ in tree networks. Algorithm $\mathcal{BA}$ is also inherently self-stabilizing and needs only $2h+2m-1$ time to broadcast m messages, where h is the height of the tree.

[MFGST05r] Nathalie Mitton, Eric Fleury, Isabelle Guérin-Lassous, Bruno Séricola, and Sébastien Tixeuil. On fast randomized colorings in sensor networks. Technical Report 1416, Laboratoire de Recherche en Informatique, Université Paris Sud, June 2005.

Abstract: We present complexity analysis for a family of self-stabilizing vertex coloring algorithms in the context of sensor networks. First, we derive theoretical results on the stabilization time when the system is synchronous. Then, we provide simulations for various schedulings and topologies. We consider both the uniform case (where all nodes are indistinguishable and execute the same code) and the non-uniform case (where nodes make use of a globally unique identifier). Overall, our results show that the actual stabilization time is much smaller than the upper bound provided by previous studies. Similarly, the height of the induced DAG is much lower than the linear dependency to the size of the color domain (that was previously announced). Finally, it appears that symmetry breaking tricks traditionally used to expedite stabilization are in fact harmful when used in networks that are not tightly synchronized.

[MFGST06c] Nathalie Mitton, Eric Fleury, Isabelle Guérin-Lassous, Bruno Séricola, and Sébastien Tixeuil. On fast randomized colorings in sensor networks. In *Proceedings of ICPADS 2006*, page to appear. IEEE Press, July 2006.

Abstract: The advent of large scale multi-hop wireless networks highlights problems of fault tolerance and scale in distributed system, motivating designs that autonomously recover from transient faults and spontaneous reconfiguration. Self-stabilization provides an elegant solution for recovering from such faults. We present complexity analysis for a family of self-stabilizing vertex coloring algorithms in the context of multi-hop wireless networks. Such "coloring" processes are used in several protocols for solving many different issues (clustering, synchronizing...). Overall, our results show that the actual stabilization time is much smaller than the upper bound provided by previous studies. Similarly, the height of the induced DAG is much lower than the linear dependency on the size of the color domain (that was previously announced). Finally, it appears that symmetry breaking tricks traditionally used to expedite stabilization are in fact harmful when used in networks that are not tightly synchronized.

[MFGST06cb] Nathalie Mitton, Eric Fleury, Isabelle Guérin-Lassous, Bruno Séricola, and Sébastien Tixeuil. Convergence dans les réseaux sans fil. In *Proceedings of Algotel 2006*, page to appear. INRIA, May 2006.

Abstract: Dans cet article, nous étudions analytiquement et par simulation le temps de convergence d'un algorithme de coloriage des nœuds dans un réseau sans fil. Nous étudions également le DAG induit et évaluons l'impact de plusieurs paramètres.

[MFGT04r] Nathalie Mitton, Eric Fleury, Isabelle Guérin-Lassous, and Sébastien Tixeuil. Self-stabilization in self-organized wireless multihop networks. Technical Report 5426, INRIA, December 2004.

URL `http://www.lri.fr/~fragile/article.php3?id_article=16`

Abstract: In large scale multihop wireless networks, flat architectures are not scalable. In order to overcome this major drawback, clusterization is introduced to support self-organization and to enable hierarchical routing. When dealing with multihop wireless networks, the robustness is a main issue due to the dynamicity of such networks. Several algorithms have been designed for the clusterization process. As far as we know, very few studies check the robustness feature of their clusterization protocols. In this paper, we show that a clusterization algorithm, that seems to present good properties of robustness, is self-stabilizing. We propose several enhancements to reduce the stabilization time and to improve stability. The use of a Directed Acyclic Graph ensures that the self-stabilizing properties always hold regardless of the underlying topology. These extra criterion are tested by simulations.

[MFGT05c] Nathalie Mitton, Eric Fleury, Isabelle Guérin-Lassous, and Sébastien Tixeuil. Self-stabilization in self-organized wireless multihop networks. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (WWAN'05)*, pages 909–915, Columbus, Ohio, USA, June 2005. IEEE Press.

Abstract: In large scale multihop wireless networks, flat architectures are not scalable. In order to overcome this major drawback, clusterization is introduced to support self-organization and to enable hierarchical routing. When dealing with multihop wireless networks, the robustness is a main issue due to the dynamicity of such networks. Several algorithms have been designed for the clusterization process. As far as we know, very few studies check the robustness feature of their clusterization protocols. In this paper, we show that a clusterization algorithm, that seems to present good properties of robustness, is self-stabilizing. We propose several enhancements to reduce the stabilization time and to improve stability. The use of a Directed Acyclic Graph ensures that the self-stabilizing properties always hold regardless of the underlying topology. These extra criterion are tested by simulations.

[MFGT05cb] Nathalie Mitton, Eric Fleury, Isabelle Guérin-Lassous, and Sébastien Tixeuil. Auto-stabilisation dans les réseaux ad hoc. In *Proceedings of Algotel 2005*, pages 45–48, May 2005.

Abstract: Dans cet article, nous choisissons un algorithme d'organisation de réseaux sans fil multi-saut, dit de *clusterisation*, et nous montrons que cet algorithme est auto-stabilisant. Nous proposons également certaines améliorations pour réduire le temps de stabilisation.

[MMPT07c] Fredrik Manne, Morten Mjelde, Laurence Pilard, and Sébastien Tixeuil. A new self-stabilizing maximal matching algorithm. In *Proceedings of the 14th International Colloquium on Structural Information and Communication Complexity (Sirocco 2007)*, page to appear. Springer Verlag, June 2007.

URL `https://hal.inria.fr/inria-00127899`

Abstract: The maximal matching problem has received considerable attention in the self-stabilizing community. Previous work has given different self-stabilizing algorithms that solves the problem for both the adversarial and fair distributed daemon, the sequential adversarial daemon, as well as the synchronous daemon. In the following we present a single self-stabilizing algorithm for this problem that unites all of these algorithms in that it stabilizes in the same number of moves as the previous best algorithms for the sequential adversarial, the distributed fair, and the synchronous daemon. In addition, the algorithm improves the previous best moves complexities for the distributed adversarial daemon from $O(n^2)$ and $O(\delta m)$ to $O(m)$ where n is the number of processes, m is the number of edges, and δ is the maximum degree in the graph.

[MMPT07r] Fredrik Manne, Morten Mjelde, Laurence Pilard, and Sébastien Tixeuil. A new self-stabilizing maximal matching algorithm. Research Report 6111, INRIA, January 2007.

URL `https://hal.inria.fr/inria-00127899`

Abstract: The maximal matching problem has received considerable attention in the self-stabilizing community. Previous work has given different self-stabilizing algorithms that solves the problem for both the adversarial and fair distributed daemon, the sequential adversarial daemon, as well as the synchronous daemon. In the following we present a single self-stabilizing algorithm for this problem that unites all of these algorithms in that it stabilizes in the same number of moves as the previous best algorithms for the sequential adversarial, the distributed fair, and the synchronous daemon. In addition, the algorithm improves the previous best moves complexities for the distributed adversarial daemon from $O(n^2)$ and $O(\delta m)$ to $O(m)$ where n is the number of processes, m is the number of edges, and δ is the maximum degree in the graph.

[MT05c] Toshimitsu Masuzawa and Sébastien Tixeuil. A self-stabilizing link coloring algorithm resilient to unbounded byzantine faults in arbitrary networks. In *Proceedings of OPODIS 2005*, Lecture Notes in Computer Science, page to appear, Pisa, Italy, December 2005. Springer-Verlag.

Abstract: Self-stabilizing protocols can tolerate any type and any number of transient faults. However, in general, self-stabilizing protocols provide no guarantee about their behavior against permanent faults. This paper proposes a self-stabilizing link-coloring protocol resilient to (permanent) Byzantine faults in arbitrary networks. The protocol assumes the central daemon, and uses $2\Delta - 1$ colors where Δ is the maximum degree in the network. This protocol guarantees that any link (u, v) between nonfaulty processes u and v is assigned a color within $2\Delta + 2$ rounds and its color remains unchanged thereafter.Our protocol is Byzantine insensitive in the sense that the subsystem of correct processes remains operating properly in spite of unbounded Byzantine faults.

[MT05r] Toshimitsu Masuzawa and Sébastien Tixeuil. A self-stabilizing link-coloring protocol resilient to unbounded byzantine faults in arbitrary networks. Technical Report 1396, Laboratoire de Recherche en Informatique, January 2005.

Abstract: Self-stabilizing protocols can tolerate any type and any number of transient faults. However, in general, self-stabilizing protocols provide no guarantee about their behavior against permanent faults. This paper proposes a self-stabilizing link-coloring protocol resilient to (permanent) Byzantine faults in arbitrary networks. The protocol assumes the central daemon, and uses $2\Delta - 1$ colors where Δ is the maximum degree in the network. This protocol guarantees that any link (u, v) between nonfaulty processes u and v is assigned a color within $2\Delta + 2$ rounds and its color remains unchanged thereafter. Thus, our protocol achieves Byzantine-fault tolerance with containment radius of one, which is trivially optimal.

[MT06ca] Toshimitsu Masuzawa and Sébastien Tixeuil. On bootstrapping topology knowledge in anonymous networks. In Ajoy K. Datta and Maria Gradinariu, editors, *Eighth International Symposium on Stabilization, Safety, and Security on Distributed Systems (SSS 2006)*, Lecture Notes in Computer Science, page to appear, Dallas, Texas, November 2006. Springer Verlag.

Abstract: In this paper, we quantify the amount of "practical" information (*i.e.* views obtained from the neighbors, colors attributed to the nodes and links) to obtain "theoretical" information (*i.e.* the local topology of the network up to distance k) in anonymous networks. In more details, we show that a coloring at distance $2k + 1$ is necessary and sufficient to obtain the local topology at distance k that includes outgoing links. This bound drops to $2k$ when outgoing links are not needed. A second contribution of this paper deals with color bootstrapping (from which local topology can be obtained using the aforementioned mechanisms). On the negative side, we show that *(i)* with a distributed daemon, it is impossible to achieve deterministic color bootstrap, even if the whole network topology can be instantaneously obtained, and *(ii)* with a central daemon, it is impossible to achieve distance m when instantaneous topology knowledge is limited to $m-1$. On the positive side, we show that *(i)* under the k-central daemon, deterministic self-stabilizing bootstrap of colors up to distance k is possible provided that k-local topology can be instantaneously obtained, and *(ii)* under the distributed daemon, probabilistic self-stabilizing bootstrap is possible for any range.

[MT06cb] Toshimitsu Masuzawa and Sébastien Tixeuil. Bounding the impact of unbounded attacks in stabilization. In Ajoy K. Datta and Maria Gradinariu, editors, *Eighth International Symposium on Stabilization, Safety, and Security on Distributed Systems (SSS 2006)*, Lecture Notes in Computer Science, page to appear, Dallas, Texas, November 2006. Springer Verlag.

Abstract: As a new challenge of containing the unbounded influence of Byzantine processes in self-stabilizing protocols, this paper introduces a novel concept of *strong stabilization*. The strong stabilization relaxes the requirement of *strict stabilization* so that processes beyond the containment radius are allowed to be disturbed by Byzantine processes, but only a limited number of times. A self-stabilizing protocol is (t, c, f)-strongly stabilizing if any process more than c hops away from any Byzantine process is disturbed at most t times in a distributed system with at most f Byzantine processes. Here c denotes the *containment radius* and t denotes the *containment times*. The possibility and the effectiveness of the strong stabilization is demonstrated using *tree orientation*. It is known that the tree orientation has no strictly stabilizing protocol with a constant containment radius. This paper first shows that the problem has no constant bound of the containment radius in a tree with two Byzantine processes even when we allow processes beyond the containment radius to be disturbed any finite number of times. Then we consider the case of a single Byzantine process and present a $(1, 0, 1)$-strongly stabilizing protocol, which achieves optimality in both containment radius and times.

[NT05r] Mikhail Nesterenko and Sébastien Tixeuil. Discovering network topology in the presence of byzantine nodes. Technical Report TR-KSU-CS-2005-1, Kent State University, May 2005.

Abstract: We present Byzantine-robust solutions to the topology discovery problem. Our programs allow each process to learn the complete topology of the network (up to the neighborhoods of the faulty nodes). The program tolerates up to a fixed number of faults. The network topology is arbitrary. The processes do *not* know either the diameter or the size of the network. The execution model is asynchronous. The processes do not use cryptographic cryptographic primitives such as digital signatures.

[NT06c] Mikhail Nesterenko and Sébastien Tixeuil. Discovering network topology in the presence of byzantine nodes. In *Prooceedings of Sirocco 2006*, page to appear, Chester, UK, July 2006. Springer Verlag.

Abstract: We study the problem of Byzantine-robust topology discovery in an arbitrary asynchronous network. We formally state the weak and strong versions of the problem. The weak version requires that either each node discovers the topology of the network or at least one node detects the presence of a faulty node. The strong version requires that each node discovers the topology regardless of faults. We focus on non-cryptographic solutions to these problems. We explore their bounds. We prove that the weak topology discovery problem is solvable only if the connectivity of the network exceeds the number of faults in the system. Similarly, we show that the strong version of the problem is solvable only if the network connectivity is more than twice the number of faults. We present solutions to both versions of the problem. Our solutions match the established graph connectivity bounds. The programs are terminating, they do not require the individual nodes to know either the diameter or the size of the network. The message complexity of both programs is low polynomial with respect to the network size.

[NT06t] Mikhail Nesterenko and Sébastien Tixeuil. Bounds on topology discovery in the presence of byzantine faults. Technical Report TR-KSU-CS-2006-01, Dept. of Computer Science, Kent State University, 2006. http://www.cs.kent.edu/techreps/TR-KSU-CS-2006-01.pdf.

Abstract: This report is a companion to another technical report to present the formal proofs that could'nt fit into the conference proceedings of this article due to space limitations. In this report we revisit the problem of Byzantine-robust topology discovery.We formally state the weak and strong versions of the problem. We focus on non-cryptographic solutions to these problems and explore their bounds. We prove that the weak topology discovery problem is solvable only if the connectivity of the network exceeds the number of faults in the system. Similarly, we show that the strong version of the problem is solvable only if the network connectivity is more than twice the number of faults.

[T00t] Sébastien Tixeuil. *Auto-stabilisation Efficace*. PhD thesis, University of Paris Sud XI, January 2000.

URL `http://tel.archives-ouvertes.fr/tel-00124843`

Abstract: When a distributed system is subject to transient failures that arbitrarily modify its state, it is crucial to recover a correct behavior within finite time. *Self-stabilization* offers such a guarantee, but usually uses large ressources. In this thesis, we focus on minimizing these ressources when such solutions exist. We introduced the concept of *transient failure detectors*, oracles that are called by processors, which notify if transient failures occurred within constant time. Our implementation enables classifying classic problems according to the specific ressources dedicated to error detection. A natural extension was to show that a local property on the local code executed by each processor is sufficient to guarantee self-stabilization of the whole system, whatever the computation assumptions and communication graph may be. Since the original algorithm is not modified, it is *overhead-free*. Similarly, we developped two *synchronizers*, that enable dynamic tasks to be solved self-stabilizingly, whose memory overhead is constant. Moreover, one of them is *snap-stabilizing*. Finally, we presented a general technique to systematically reduce the communication cost, assuming a *bounded retransmission delay*, and we gave a general framework and tools to design self-stabilizing algorithms in this context.

[T01c] Sébastien Tixeuil. On a space-optimal distributed traversal algorithm. In Springer Verlag, editor, *Fifth Workshop on Self-stabilizing Systems (WSS'2001)*, volume LNCS 2194, pages 216–228, Lisbon, Portugal, October 2001.

Abstract: A traversal algorithm is a systematic procedure for exploring a graph by examining all of its vertices and edges. A traversal is Eulerian if every edge is examined exactly once. We present a simple deterministic distributed algorithm for the Eulerian traversal problem that is space-optimal: each node has exactly d states, where d is the outgoing degree of the node, yet may require $O(m^2)$ message exchanges before it performs an Eulerian traversal, where m is the total number of edges in the network. In addition, our solution has failure tolerance properties: *(i)* messages that are exchanged may have their contents corrupted during the execution of the algorithm, and *(ii)* the initial state of the nodes may be arbitrary. Then we discuss applications of this algorithm in the context of self-stabilizing virtual circuit construction and cut-through routing. Self-stabilization guarantees that a system eventually satisfies its specification, regardless of the initial configuration of the system. In the cut-through routing scheme, a message must be forwarded by intermediate nodes before it has been received in its entirety. We propose a transformation of our algorithm by means of randomization so that the resulting protocol is self-stabilizing for the virtual circuit construction specification. Unlike several previous self-stabilizing virtual circuit construction algorithms, our approach has a small memory footprint, does not require central preprocessing or identifiers, and is compatible with cut-through routing.

[T06bc] Sébastien Tixeuil. *Réseaux mobiles ad hoc et réseaux de capteurs sans fils*, chapter Algorithmique répartie tolérante aux pannes dans les systèmes à grande échelle, pages 251–284. Information, Commande, Communication. Lavoisier, March 2006.

[T06r] Sébastien Tixeuil. Vers l'auto-stabilisation des systèmes à grande echelle. Habilitation à diriger les recherches, Université Paris-Sud XI, Orsay, France, May 2006.

URL `http://tel.archives-ouvertes.fr/tel-00124848`

[T07c] Sébastien Tixeuil. *Wireless Ad Hoc and Sensor Networks*, chapter Fault-tolerant distributed algorithms for scalable systems. ISTE, October 2007. ISBN: 978 1 905209 86.

URL `http://www.iste.co.uk/index.php?p=a&ACTION=View&id=166`

[T95m] Sébastien Tixeuil. Auto-stabilisation par propagation de compteur. Master's thesis, Université de Paris Sud, June 1995.

Abstract: En partant d'une solution récente donnée par G. Varghese à un problème d'auto-stabilisation sur un anneau, dans un modèle à passage de messages, nous étudions les possibilités de réaliser effectivement cette solution dans le cadre du *Token Ring*. Nous présentons une solution originale valable dans ce contexte et nous analysons le surcoût qu'elle apporte par rapport à l'algorithme classique du *Token Ring*. Après examen, il apparait que ce surcoût est extrèmement minime. Ceci laisse entrevoir la possibilité d'une utilisation pratique des solutions auto-stabilisantes.

[TB96c] Sébastien Tixeuil and Joffroy Beauquier. Self-stabilizing token ring. In *Proceedings of the Eleventh International Conference on System Engineering (ICSE'96)*, 1996.

Abstract: Self-stabilization is a promising technique for making network protocols resisting to any memory and channel corruptions. Primarly, we present a self-stabilizing RAR Token Ring (SSTR) protocol derived from a self-stabilizing mutual exclusion protocol through an automatic transformation we provide. Secondly, we enhance Dijkstra's classic self-stabilizing mutual exclusion protocol in terms of forwarding delay, and use it as the underlying self-stabilizing mutual exclusion protocol for the SSTR algorithm. We believe our solution can compete with actual Token Ring protocols when corruptions do exist but are rare.

[TB96r] Sébastien Tixeuil and Joffroy Beauquier. Self-stabilizing token ring. Technical Report 1033, Laboratoire de Recherche en Informatique, February 1996.

Abstract: Self-stabilization is a promising technique for making network protocols resisting to any memory and channel corruptions. Primarly, we present a self-stabilizing RAR Token Ring (SSTR) protocol derived from a self-stabilizing mutual exclusion protocol through an automatic transformation we provide. Secondly, we enhance Dijkstra's classic self-stabilizing mutual exclusion protocol in terms of forwarding delay, and use it as the underlying self-stabilizing mutual exclusion protocol for the SSTR algorithm. We believe our solution can compete with actual Token Ring protocols when corruptions do exist but are rare.

[THS06c] Sébastien Tixeuil, William Hoarau, and Luis Silva. An overview of existing tools for fault-injection and dependability benchmarking in grids. In *Second CoreGRID Workshop on Grid and Peer to Peer Systems Architecture*, Paris, France, January 2006.

Abstract: In this paper we review several existing tools for fault injection and dependability benchmarking in grids. We emphasis on the FAIL-FCI faultinjection software that has been developed in INRIA Grand Large, and a benchmark tool called QUAKE that has been developed in the University of Coimbra. We present the state-of-the-art and we explain the importance of these tools for dependability assessment of Grid-based applications and Grid middleware.

[TSHJBT05c] Sébastien Tixeuil, Luis Moura Silva, William Hoarau, Gonçalo Jesus, Jo ao Bento, and Frederico Telles. Fault-injection and dependability benchmarking for grid computing middleware. In *Proceedings of CoreGrid Integration Workshop*, page to appear, November 2005.

Abstract: In this paper we will present some work on dependability benchmarking for Grid Computing that represents a common view between two groups of Core-Grid: INRIA-Grand Large and University of Coimbra. We present a brief overview of the state of the art, followed by a presentation of the FAIL-FCI system from INRIA that provides a tool for fault-injection in large distributed systems. Then we present DBGS, a dependable Benchmark for Grid Services and we present some experimental results. We conclude the paper with some considerations about the avenues of research ahead that both groups would like to contribute, on behalf of the Core-GRID network.