# 高信頼分散システムのためのグループ通信とその故障発見方法

# Group Communication and its Failure Detection Method for Dependable Distributed Systems

（研究課題名：大規模分散アルゴリズム開発及び性能評価のツール構築）

「機能と構成」領域　Xavier DÉFAGO

## 要　　旨 (Outline)

Our objective is to pave the ground for designing more efficient fault-tolerant mechanisms for distributed systems, in order to widen their use. Concretely, this means gathering a better understanding of the various performance tradeoffs involved in complex fault-tolerant distributed protocols, and designing mechanisms that are better adapted to their environment.

Group communication provides a common abstraction for building complex fault-tolerant distributed systems. These systems use group communication primitives as a means to maintain a consistent behavior between several running programs. While many group communication mechanisms have been proposed in the literature, little is know about their respective performance and scalability. Since most mechanisms have their own advantages and drawbacks, choosing one over another depend on many parameters. We have thus extensively studied performance tradeoffs of group communication mechanisms in failure-free systems.

To tolerate failures, group communication mechanisms rely on failure detection, and the performance of the former is heavily determined by the performance of the latter. We have studied failure detection mechanisms and proposed a novel method to integrate failure detection in a larger system. Our future aim is to use this method to eventually provide a fully generic failure detection service for distributed systems.

## 1．研究のねらい (Research objective)

The objective of the research is to better understand the performance tradeoffs associated with fault-tolerant mechanisms for distributed systems. In particular, group communication protocols, such as Total Order Broadcast, are key factors in determining the performance of the system in the absence of failures. While failure-free executions constitute the common case, the occurrence of failures should not affect system performance too drastically, or else failures risk being perceived by the users, thus defeating the objective of masking them. The performance in the face of failures depends mostly on the ability of the system to detect failures promptly and accurately, but this is made difficult by an inherent tradeoff between these two measures. Thus the second objective is to provide a generic failure detection service, the speed and accuracy of

which can be best tuned to the specific needs of each part of the entire distributed system.

## 2. 研究方法と成果 (Research approach and results)

In this research, we have made three major contributions.

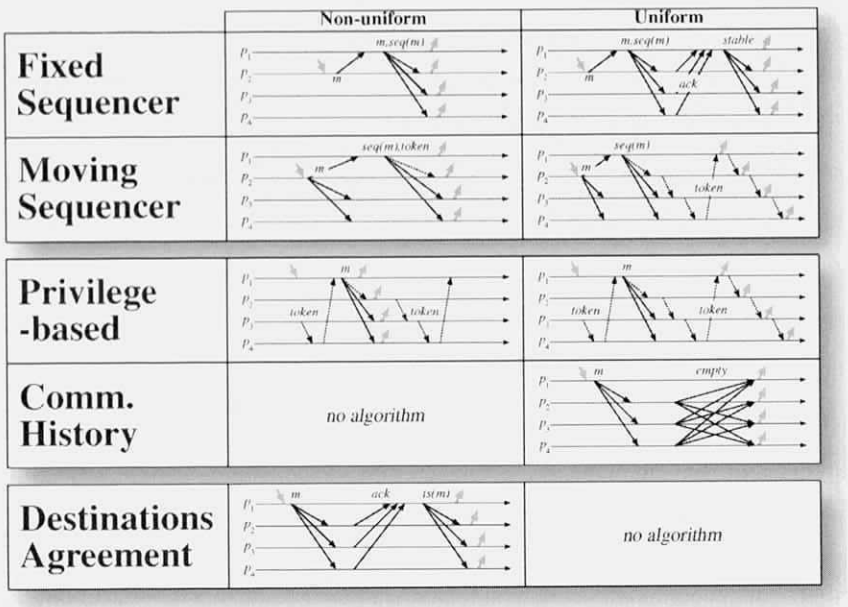
Firstly, we have studied several group communication protocols, with a particular focus on the problem of Total Order Broadcast (also called Atomic Broadcast) because it is an important component for many kinds of practical systems, including distributed databases, distributed shared memory, highly-available replicated services, etc.

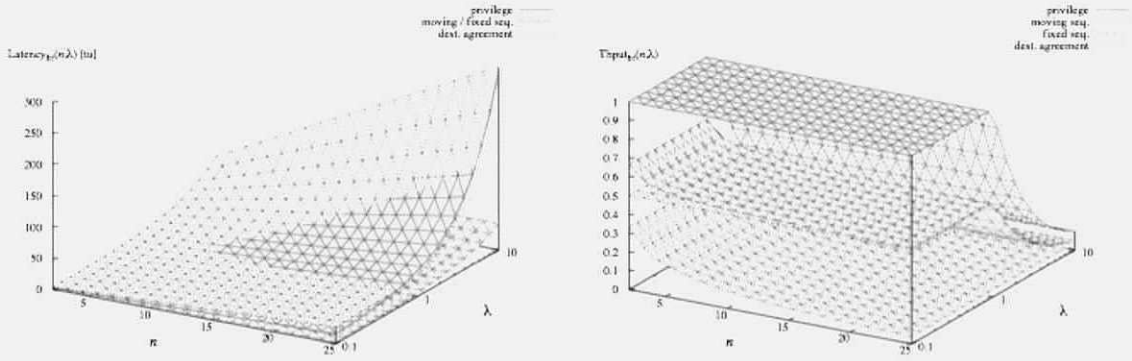
Secondly, we have proposed a novel failure detection method, called accrual failure detectors, with the ultimate goal of providing failure detection as a generic and highly-configurable service for distributed systems. Studying failure detection is also a prerequisite to understand the actual performance of group communication protocols in the face of failures.

Thirdly, we have developed a communication platform to provide a better support for the evaluation of distributed algorithms.

### 2.1 Group Communication and Distributed Agreement

Group communication and distributed agreement are basic primitives to maintain the cohesion of a distributed sy: ›mmon issues. A very practical inst: lled *Total Order Broadcast* (also Atomic Broadcast). In short, this primitive allow any of the processes to broadcast messages any time, but guarantees that all destinations will always see (and process) the messages in the same exact order. Among other things, Total Order Broadcast is a key component for supporting the replication of running programs and services (e.g., web or Grid services).






Indeed, assuming that all replicas have the same initial state (i.e., value of variables, etc.), then Total Order Broadcast is used to issue requests to the service. Because of the guarantees offered by the primitive, all replicas perform the same actions in the same sequence, and thus their state change in exactly the same way. The benefit is that the replicas remain exact copies of each others, and thus the service can remain operational even after the crash of some of the replicas, that is, provided that the Total Order Broadcast can actually tolerate the crash of some of the processes.

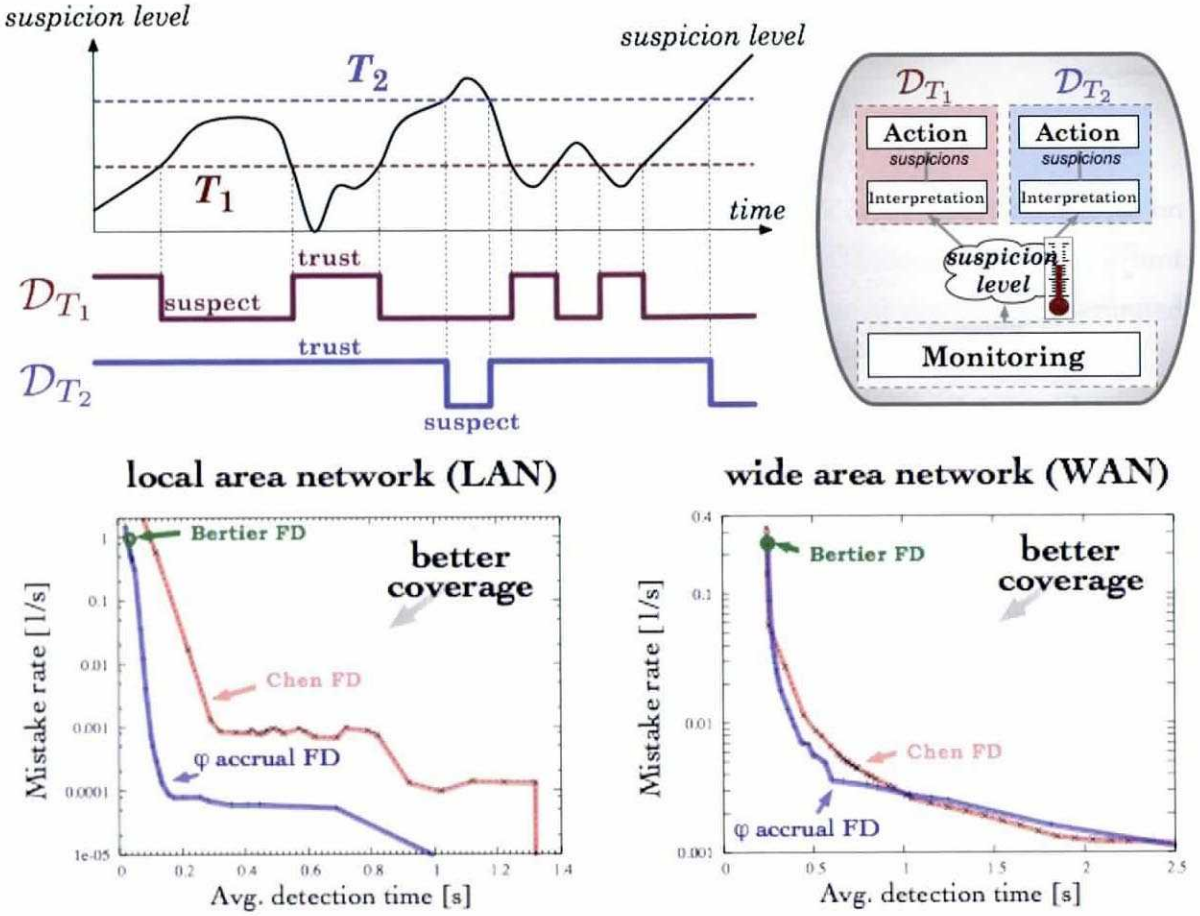
There exist many algorithms to solve Total Order Broadcast, most of which can tolerate failures. However, they do not offer exactly the same guarantees, and their respective performance can vary drastically. We have thus surveyed and analyzed about sixty different Total Order Broadcast algorithms [1], and identified five basic families and a total of thirteen subclasses. The basic families, defined on the decision process used to generate the delivery order, are called *fixed sequencer*, *moving sequencer*, *privilege-based*, *communication history*, and *destinations agreement*.

Using this classification, we have defined representative algorithms for each of the class, and compared their performance and scalability [3] in different network environment. We have also defined a novel replication technique using a variant of a destination agreement Total Order Broadcast algorithm [2].

## 2.2 Accrual Failure Detectors

Failure detection plays an essential role in ensuring fault tolerance in distributed systems. Recently, many people have come to realize that failure detection ought to be provided as some form of generic service.

The performance of fault-tolerant distributed systems, and their ability to mask failures from the viewpoint of their users, depend greatly on the characteristics of the failure detection. This is especially true when it comes to group communication algorithms, such as the Total Order Broadcast discussed in the previous. When considering a complete system, it turns out that




failure detection is required at many different levels, but often with very different performance requirements. Roughly speaking, the performance of a failure detector can be expressed by two measures: *detection latency* (i.e., how long it takes until a real failure is detected) and *accuracy* (i.e., how often a running process is erroneously suspected).

In conventional systems, there is always a tradeoff between detection latency (also called detection time) and accuracy, but different operations will benefit from very different settings. For instance, many Total Order Broadcast algorithms will have best overall efficiency with a failure detection latency in the order of a hundred milliseconds, even if the detection is often erroneous. In contrast, a global system reconfiguration will benefit from more accurate detection, even if this leads to a considerable detection latency. It is thus difficult to provide a failure detection scheme that simultaneously provides ideal performance for both.

To address this problem, we have developed the notion of accrual failure detector, promoting a clean decomposition of roles and formally establishing the link with the basic theory on failure detection [4]. Instead of a binary value (trust or suspect), accrual failure detectors associate to each process a real value representing a *suspicion level*. By establishing the clear link with the theory on failure detection, we have identified minimal properties whereby the failure detection scheme can be used for solving distributed agreement problems, such as Total Order Broadcast and Consensus.

Broadcast and Consensus.

We have developed several implementations of accrual detectors and, in particular, a highly adaptive one called φ accrual failure detector [7]. We have conducted extensive performance measurements, comparing the performance of our φ accrual failure detector with that of other state-of-the-art failure detection schemes. Our results have shown that, for the same latency, our scheme could perform up to ten times more accurately in a local network. Our experiments on an intercontinental network (between Japan and Switzerland) have shown comparable performance in spite of the change of interaction scheme, thus effectively showing the practicality of our approach.

## 2.3 NekoLS Prototyping Platform

To conduct our experiments more efficiently, we have developed a communication platform called NekoLS. This platform is an extension of an earlier system of us, called Neko, that allowed an easier development of distributed algorithms, and with which the same code can be executed either in a real network environment or on a single machine, within a simulated network. We have made many improvements of this system, the most notable of which is a seamless integration with the SSFNet project; a network simulator aimed for describing large and complex network topologies.

For describing algorithms, the Neko platform is based on a simple layered architecture. While this choice was good for describing simple protocols, it turned out that even moderately complex protocols were very difficult to design elegantly, due for a large part to the difficulty to prevent deadlocks in the protocol. To address this issue, we have been working on a novel mechanism for the composition of micro-protocols.

Except for the most basic examples, describing distributed algorithms is a very complex task, as this often requires developing some basic functionalities, such as retransmission, flow control, FIFO or causal order preservation, etc. Similar to object systems for conventional software, the ideal of distributed systems engineering is to allow the reuse of basic protocol components, called micro-protocols. Unfortunately, the interactions between even simple micro-protocols can yield very complex behaviors that are not always those desired. So, instead of a rigid layered structure, we are developing a more flexible architecture that allow the composition of micro-protocols while avoiding (or at least reducing) undesirable side-effects such as those resulting from concurrency (e.g., deadlocks, race conditions).

## 3. 今後の展望 (Future plans)

In the future of this research, we will be using the concept of accrual failure detectors to design a fully generic failure detection service for distributed systems. This will make it much eas-

ier for common programmers to develop fault-tolerant distributed applications, thus avoiding the current plague of hard-coded timeouts that result in systems with very unstable behavior, or the explosion of redundant control messages generated by different applications. Later, we will try to extend this failure detection service to provide a more general monitoring infrastructure for distributed systems, and we will focus particularly on novel environments, such as sensor networks and autonomous mobile systems.

Based on our work on group communication, we will develop a communication middleware to support the fault-tolerant coordination and cooperation of groups of mobile robots. In particular, we have found that Total Order Broadcast provides a very adequate and pragmatic building block on which we can construct such a middleware.

## 4. 成果リスト (List of results)

### 招待講演 (Special lectures)

2005年12月5日 "*Failure Detection in Distributed Systems: Retrospective and recent advances.*" Tutorial. 6th Intl. Conf. on Parallel and Distributed Computing, Applications and Technologies.

2005年7月2日 "*Revisiting Failure Detection for Grid Systems.*." Invited talk. 48th meeting IFIP working group 10.4 (dependable computing & fault-tolerance).

2004年10月17日 "*Panel: Dependable replicated data: Strategies, drawbacks and benchmarking.*" Panelist. Workshop on Dependable Distributed Data Management.

2003年7月3日 "*Fault-Tolerant Group Communication and the Many Faces of Scalability.*" Invited lecture. Information and Communications University (ICU).

2003年7月3日 "*Fault-Tolerant Group Communication and the Many Faces of Scalability.*" Invited lecture. Electronics and Telecommunications Research Institute (ETRI).

### 論　文 (Publications)

#### international journals

[1] X. Défago, A. Schiper, and P. Urbán. Total order broadcast and multicast algorithms: Taxonomy and survey. *ACM Computing Surveys*, 36(4):372-421, December 2004. ACM Press.

[2] X. Défago and A. Schiper. Semi-passive replication and Lazy Consensus. *Journal of Parallel and Distributed Computing*, 64(12):1380-1398, December 2004. Elsevier.

[3] X. Défago, A. Schiper, and P. Urbán. Comparative performance analysis of ordering strategies in atomic broadcast algorithms. *IEICE Trans. on Information and Systems*, Vol.E86-D, No.12, pp.2698-2709, December 2003.

#### international conferences (refereed)

[4] X. Défago, P. Urbán, N. Hayashibara, T. Katayama. Definition and specification of accrual failure detectors. In *Proc. IEEE/IFIP Intl. Conf. on Dependable Systems and Networks,* pp. 206-215, June

2005. IEEE CS Press.

[5] N. Hayashibara, X. Défago, M. Takizawa, and T. Katayama. Information propagation on the $\phi$ failure detector. In *Proc. 16th Intl. Workshop on Database and Expert Systems Applications*, pp.72–76, August 2005.. IEEE CS Press.

[6] R. Yared, X. Défago, and T. Katayama. Fault-tolerant group membership protocols using physical robot messengers. In *Proc. 19th IEEE Intl. Conf. on Advanced Information Networking and Applications*, Vol.1, pp.921–926, March 2005. IEEE CS Press.

[7] N. Hayashibara, X. Défago, R. Yared, and T. Katayama. The $\phi$ accrual failure detector. In *Proc. 23rd IEEE Intl. Symp. on Reliable Distributed Systems*, pp. 66-78, October 2004. IEEE CS Press.

[8] A. Ben Hassine, X. Défago, and T. B. Ho. Agent-based approach to dynamic meeting scheduling problems. In *Proc. 3rd Intl. Joint Conf. on Autonomous Agents and Multi Agent Systems*, Vol.3, pp.1130–1137, July 2004. IEEE CS Press.

[9] K. Satou, Y. Nakajima, S. Tsuji, X. Défago, and A. Konagaya. An integrated system for distributed bioinformatics environment on grids. In *Grid Computing in Life Science: First Intl. Life Science Grid Workshop*, May 2004. LNCS 3370/2005, Springer.

[10] S. Souissi, X. Défago, and T. Katayama. Decomposition of fundamental problems for cooperative autonomous mobile systems. In *Proc. 24th IEEE Intl. Conf. on Distributed Computing Systems Workshops*, pp.554-560, March 2004. IEEE CS Press.

[11] J. C. Clemente Litrán, X. Défago, and K. Satou. Asynchronous peer-to-peer communication for failure resilient distributed genetic algorithms. In *Proc. 15th IASTED Intl. Conf. on Parallel and Distributed Computing and Systems*, Vol.II, pp.769–773, November 2003.

[12] X. Défago, N. Hayashibara, and T. Katayama. On the design of a failure detection service for large scale distributed systems. In *Proc. Intl. Symp. Towards Peta-Bit Ultra-Networks*, pp.88–95, September 2003.

[13] M. Wiesmann, X. Défago, and A. Schiper. Group communication based on standard interfaces. In *Proc. 2nd IEEE Intl. Symp. on Network Computing and Applications*, pp.140-147, April 2003.

...and a few other papers.

## international conferences (invited, short, position papers)

[14] E. Anceaume, X. Défago, M. Gradinariu, and M. Roy. Brief Announcement: Towards a theory of self-organization, In *Proc. 19th Intl. Symp. on Distributed Computing*, LNCS 3724, pp. 505–506, September 2005. Springer-Verlag.

[15] X. Défago. Semi-passive replication and the eventual leadership (invited paper). In *Proc. Workshop on Dependable Distributed Data Management*, pp.13–18, October 2004.

[16] N. Hayashibara, X. Défago, T. Katayama. Two-ways adaptive failure detection with the $\phi$-failure detector. In *Proc. Intl. Workshop on Adaptive Distributed Systems*, pp.22–27, October 2003.

## local conferences

[17] S. Souissi, X. Défago, and T. Katayama. Convergence of a uniform circle formation algorithm for distributed autonomous mobile robots. In *Proc. Japan-Tunisia Workshop on Computer Systems and Information Technology*, July 2004.

[18] X. Défago, N. Hayashibara, and T. Katayama. An adaptive failure detection service for large-scale distributed systems. In *Actes Journées Scientifiques Francophones*, November 2003.

[19] A. Ben Hassine, X. Défago, and T. B. Ho. Novel approach for the dynamic resolution of meeting scheduling problem. In *Actes Journées Scientifiques Francophones*, November 2003.

[20] S. Souissi, X. Défago, and T. Katayama. Specification of recurrent problems in distributed cooperative mobile robotics. In *Actes Journées Scientifiques Francophones*, November 2003.

[21] P. Urbán, X. Défago, and T. Katayama. NekoLS: prototyping and simulation of large-scale distributed systems. In *Actes Journées Scientifiques Francophones*, November 2003.