Brief Announcement: From an Intermittent Rotating Star to a Leader

Antonio Fernández
Universidad Rey Juan Carlos, Móstoles, Spain

Michel Raynal
IRISA, Campus Beaulieu, Rennes, France

Categories and Subject Descriptors C.2.4 [**Computer-Communication Network**]: Distributed Systems D.4.5 [**Operating Systems**] Reliability -*fault-tolerance*;

General Terms: Algorithms, Reliability, Theory

Keywords: Asynchrony, Eventual leader, Fault-tolerance.

1. THE EVENTUAL LEADER ORACLES

The class of eventual leader oracle [2] (usually denoted Ω) provides the processes with a *leader* primitive that outputs a process id each time it is called, and such that, after some finite but unknown time, all its invocations return the same id, which is the identity of a correct process (a process that does not commit failures). Such an oracle is particularly weak: (1) a correct leader is eventually elected, but there is no knowledge on when it is elected; and (2) several (correct or not) leaders can co-exist before a single correct leader is elected.

Ω has two fundamental features. The first lies in the fact that, despite its very weak definition, it is the weakest oracle (among the oracles whose output is limited to information on failures) that allows to solve the consensus problem [2], which is one of the most fundamental problems of fault-tolerant distributed computing. The second main feature lies in the fact that it allows the design of *indulgent* protocols [4], i.e., protocols that never violate their safety property, whatever the (incorrect) behavior of the underlying oracle [4] (they can then lose only their liveness).

Unfortunately, Ω cannot be implemented in pure asynchronous distributed systems prone to process crashes. (The very existence of such an implementation would contradict the impossibility of solving consensus in such systems.) A main challenge of asynchronous fault-tolerant distributed computing is consequently to identify properties that are at the same time "weak enough" in order to be "nearly always" satisfied by the underlying asynchronous system, while being "strong enough" to allow Ω to be implemented during the "long periods" in which they are satisfied.

2. IMPLEMENTING Ω

Up to now two main approaches have been investigated to implement Ω in crash-prone asynchronous distributed systems. Both consist on enriching the asynchronous system with additional assumptions. These two approaches are orthogonal, namely, one is related to timing assumptions, the other is related to message pattern assumptions.

The eventual timely link approach. The first approach considers that the asynchronous system eventually satisfies additional *synchrony* properties. Considering a reliable communication network, the very first papers (e.g., [6]) assumed that all the links are *eventually timely*. This assumption means that there is a time τ_0 after which there is a bound δ -possibly unknown- such that, for any time $\tau \geq \tau_0$, a message sent at time τ is received by time $\tau + \delta$. This approach has then been refined to obtain weaker assumptions. It has been shown in [1] that Ω can be built as soon as there is a correct process that has only t eventually timely links (where t is an upper bound on the number of processes that can crash); such a process is called an *eventual t-source*. (Let us notice that, after the receiver has crashed, the link from a correct process to a crashed process is always timely).

Another time-based assumption has been proposed in [7] where the notion of *eventual t-accessibility* is introduced. A process p is eventual t-accessible if there is a time τ_0 such that, at any time $\tau \geq \tau_0$, there is a set $Q(\tau)$ of t processes such that $p \notin Q(\tau)$ and a message broadcast by p at τ receives a response from each process of $Q(\tau)$ by time $\tau + \delta$ (where δ is a bound known by the processes). The very important point here is that the set $Q(\tau)$ of processes whose responses have to be received in a timely manner is not fixed and can be different at distinct times.

The notions of t-source and t-accessibility cannot be compared (which means that none of them can be simulated from the other). In a very interesting way these two notions have been combined in [5] where is defined the notion of *eventual t-moving source*. A process p is an eventual t-moving source if there is a time τ_0 such that at any time $\tau \geq \tau_0$ there is a set $Q(\tau)$ of t processes such that $p \notin Q(\tau)$ and a message broadcast by p at τ is received by each process in $Q(\tau)$ by time $\tau + \delta$. As we can see, the *eventual t-moving source* assumption is weaker than the *eventual t-source* as the set $Q(\tau)$ can vary with τ.

The message pattern approach. A totally different approach to build Ω has been introduced in [8]. That approach does not rely on timing assumptions and timeouts. It states a property on the *message exchange pattern* that, when satisfied, allows Ω to be implemented. The statement of such a property involves the system parameters n and t.

Let us assume that each process broadcasts queries and, for each query, waits for the corresponding responses. Given

PODC'07, August 12–15, 2007, Portland, Oregon, USA.
ACM 978-1-59593-616-5/07/0008.

a query, a response that belongs to the first $(n-t)$ responses to that query is said to be a *winning* response. Otherwise, the response is a *losing* response (then, that response is slow, lost or has never been sent because its sender has crashed). It is shown that Ω can be built as soon as the following behavioral property is satisfied: "There are a correct process p and a set Q of t processes such that $p \notin Q$ and eventually the response of p to each query issued by any $q \in Q$ is always a winning response (until -possibly- the crash of q)." When $t = 1$, this property becomes: "There is a link connecting two processes that is never the slowest (in terms of transfer delay) among all the links connecting these two processes to the rest of the system." A probabilistic analysis for the case $t = 1$ shows that such a behavioral property on the message exchange pattern is practically always satisfied [8].

While the *message pattern* and the *eventual timely link* approaches cannot be compared, they can be combined. This combination [9] shows that Ω can be implemented as soon as there is a correct process p and a time τ_0 after which there is a set Q of t processes q such that $p \notin Q$ and either (1) each time a process $q \in Q$ broadcasts a query, it receives a winning response from p, or (2) the link from p to q is timely. As it can be seen, if only (1) is satisfied, we obtain the *message pattern* assumption, while, if only (2) is satisfied, we obtain the *eventual t-source* assumption. More generally, here, the important fact is that the message pattern assumption and the timely link assumption are combined at the "finest possible" granularity level, namely, the link level.

3. TOWARDS WEAKER AND WEAKER SYNCHRONY ASSUMPTIONS

A quest for a fault-tolerant distributed computing holy grail is looking for the *weakest synchrony assumptions* that allow implementing Ω. Differently from the quest for the weakest information on failures that allows solving the consensus problem (whose result was Ω [2]), it is possible that this quest be endless. This is because we can envisage lots of base asynchronous computation models, and enrich each of them with appropriate assumptions that allow implementing Ω in the corresponding system. Such a quest should be based on a well-formalized definition of a low level asynchronous model, including all the models in which Ω can be implemented. There is no guarantee that such a common base model exists.

This paper (see [3] for a full version) is a step in that direction. It considers the classical asynchronous computing model where processes can crash. They communicate through a reliable network. The paper shows that it is possible to implement Ω in an asynchronous system from a synchrony assumption weaker than any of the previous ones, namely, *eventual t-source*, *eventual t-moving source*, or the *message pattern* assumption. Interestingly, these specific assumptions become particular cases of the more general (and weaker) assumption that is proposed. In that sense, the paper not only proposes a weaker assumption, but has also a generic dimension.

The proposed behavioral assumption (that we denote $\mathcal{A}$) assumes that each process regularly broadcasts ALIVE(rn) messages, where rn is an increasing round number (this can always be done in an asynchronous system). The sending of ALIVE(rn) messages by the processes can be seen as an *asynchronous round*, each round number defining a new round.

To make easier the presentation we describe first an assumption $\mathcal{A}^+$ of which $\mathcal{A}$ is a weakening. $\mathcal{A}^+$ is as follows. There is a correct process p and a round number RN_0 such that, for each $rn \geq RN_0$, there is a set $Q(rn)$ of t processes such that $p \notin Q(rn)$ and for each process $q \in Q(rn)$ either (1) q has crashed, or (2) the message ALIVE(rn) sent by p is received by q at most δ time units after it has been sent (the corresponding bound δ can be unknown), or (3) the the message ALIVE(rn) sent by p is received by q among the first $(n-t)$ ALIVE(rn) messages received by q (i.e., it is a winning message among ALIVE(rn) messages received by q). It is easy to see, that if only (1) and (2) are satisfied, $\mathcal{A}^+$ boils down to the eventual t-moving source assumption, while if only (1) and (3) are satisfied, it boils down to a *moving* version of the message pattern assumption (because the set $Q()$ can change over time). The set of processes $\{p\} \cup Q(rn)$ defines a star centered at p. As it must have at least t points (links), we say it is a *t-star*. Moreover, as $Q(rn)$ can change at each round number, we say that p is the *center* of an *eventual rotating t-star* ("eventual" because there is an arbitrary finite number of round numbers during which the requirement can be not satisfied).

While $\mathcal{A}^+$ allows implementing Ω, it appears that a weakened form of that assumption is sufficient. This is the assumption $\mathcal{A}$. It is sufficient that p be the center of an eventual rotating t-star only for a subset of the round numbers. More precisely, $\mathcal{A}$ requires that there is an infinite sequence $S = s_1, s_2, \ldots$ of (not necessarily consecutive) round numbers, and a bound D (not necessarily known), such that, $\forall k \geq 1$, $s_{k+1} - s_k \leq D$, and there is a process p that is the center of a rotating t-star when we consider only the round numbers in S. We call such a configuration an *eventual intermittent rotating t-star* The interested reader will find more details on the $\mathcal{A}^+$ and $\mathcal{A}$ assumptions, and corresponding Ω algorithms in [3].

4. REFERENCES

[1] Aguilera M.K., Delporte-Gallet C., Fauconnier H. and Toueg S., Communication Efficient Leader Election and Consensus with Limited Link Synchrony. *23th ACM PODC*, pp. 328-337, 2004.

[2] Chandra T.D., Hadzilacos V. and Toueg S., The Weakest Failure Detector for Solving Consensus. *Journal of the ACM*, 43(4):685-722, 1996.

[3] Fernández A. and Raynal M., ¿From an intermittent rotating star to a leader. *Tech Report 1810*, IRISA, Universitè de Rennes, France, 2006.

[4] Guerraoui R., Indulgent Algorithms. *19th ACM PODC*, pp. 289-298, 2000.

[5] Hutle M., Malkhi D., Schmid U. and Zhou L., Chasing the weakest system model for implementing Ω and consensus. *BA, Proc. 8th Symp. SSS*, LNCS #4280, pp. 576-577, 2006.

[6] Larrea M., Fernández A. and Arévalo S., Optimal Implementation of the Weakest Failure Detector for Solving Consensus. *Proc. 19th IEEE SRDS*, pp. 52-60, 2000.

[7] Malkhi D., Oprea F. and Zhou L., Ω Meets Paxos: Leader Election and Stability without Eventual Timely Links. *Proc. 19th DISC*, LNCS #3724, pp. 199-213, 2005.

[8] Mostéfaoui A., Mourgaya E., and Raynal M., Asynchronous Implementation of Failure Detectors. *Proc. Int'l IEEE Conf. on Dependable Systems Networks (DSN'03)*, pp. 351-360, 2003.

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