# Brief Announcement: Chasing the Weakest System Model for Implementing $\Omega$ and Consensus

Martin Hutle[1,3], Dahlia Malkhi[2], Ulrich Schmid[3], and Lidong Zhou[2]

[1] Ecole Polytechnique Fédérale de Lausanne (EPFL)
[2] Microsoft Research
[3] Vienna University of Technology, Embedded Computing Systems Group 182-2

The chase for the weakest system model that allows to solve consensus has long been an active branch of research in distributed algorithms. To circumvent the FLP impossibility in asynchronous systems, many models in between synchrony and asynchrony have been proposed over the years. Of specific interest is the chase for the weakest system model that allows the implementation of an eventual leader oracle $\Omega$, and thus also enables consensus to be solved.

Recently, Aguilera et al. [ADGFT04] and Malkhi et al. [MOZ05] presented two system models which are weaker than all previously proposed models where $\Omega$ can be implemented. The former model assumes unicast steps and at least one correct process with $f$ outgoing eventually timely links. The latter assumes broadcast steps and at least one correct process with $f$ bidirectional but moving eventually timely links. Consequently, those models are incomparable.

Our main result in the full paper [HMSZ05:TR] shows that $\Omega$ can be implemented in a system with at least one process with $f$ outgoing moving eventually timely links, assuming either unicast or broadcast steps. Our construction seems to solve consensus (via $\Omega$) in the weakest system model known so far.

**Definition 1 (The weak model $\mathcal{S}^{\rightarrow}_{f*}$.).**
*Informally, a $\diamond$moving-$f$-source is a correct process that, eventually, if it sends a message to all other processes at time $t$, at least $f$ of these messages are timely. Our system $\mathcal{S}^{\rightarrow}_{f*}$ assumes the existence of at least one $\diamond$moving-$f$-source. All other links can be totally asynchronous.*

**Theorem 1.** *It is possible to implement $\Omega$ in system $\mathcal{S}^{\rightarrow}_{f*}$.*

We also provide matching lower bounds for the communication complexity in this model, which are based on an interesting "stabilization property" of infinite runs. Those results reveal a price to be paid for the relaxation of synchrony properties, compared, e.g., with the last algorithm in Aguilera et al. [ADGFT04] where only $f$ links are required to carry messages forever. Thus, these results indicate an interesting tradeoff between synchrony assumptions and communication complexity.

**Theorem 2.** *For all $n > f + 1 \geq 2$, in a system $\mathcal{S}^{\rightarrow}_{f*}$ with reliable links and $n$ processes where up to $f$ processes may crash, any implementation of $\Omega$ requires at least $\frac{nf}{2}$ links to carry messages forever in some run. This holds even when every process is a perpetual moving-$f$-source, and $\delta$ is known.*

In the full paper [HMSZ05:TR] we give an algorithm that matches the $\Omega(nf)$ lower bound, i.e., where only $O(nf)$ links carry messages forever.

*The Algorithm for $\mathcal{S}_{f*}^{\rightarrow}$.* We now provide an informal description of the main ingredients of our solution. The algorithm bears similarities to the algorithm of [ADGFT04], with the following important distinctions: It introduces suspicion sequence-numbers, and the agreement on suspicions is done on a per-sequence-number basis.

The algorithm works as follows: Every process $p$ periodically sends Alive messages with increasing sequence numbers ($seq_p$) to all. Every receiver process $q$ maintains a receiver-sequence number ($rseq_q$), and expects to receive an Alive message with a sequence number matching $rseq_q$ from every other process $p$ within a timeout period. A timer is used for terminating the wait; both $rseq_q$ and the timeout value are incremented when the timer expires.

Every receiver process $q$ maintains an array $counter_q[p]$, which essentially contains the number of suspicions of sender $p$ encountered at $q$ so far: The sender $p$ is suspected at $q$ if $q$ is notified of the fact that at least $n - f$ receivers experienced a timeout for the same sequence number $s$. This notification is done via Suspect messages, which are sent to all by any receiver process that experienced a timeout for sender $p$ with sequence number $s$. In addition, counter values are piggybacked onto Alive messages. If a larger counter value for process $p$ is observed in any Alive message, $counter_q[p]$ adopts this value. The process $p = \ell$ with minimal counter value in $counter_q[p]$ (or the minimal process id in case of several such entries) is elected as $q$'s leader.

Informally, the correctness of the algorithm follows from the following reasoning: At the time the $\diamond$moving-$f$-source becomes a moving-$f$-source, at least $f$ outgoing links of the source $p$ carry timely messages at any time. Thus, eventually, it is impossible that the quorum of $n - f$ Suspect messages is reached for $p$ for any sequence number. Note that this even holds true if some of the $f$ timely receiver processes have crashed. Consequently, all processes stop increasing the counter for process $p$, whereas the counter of every crashed sender process keeps increasing forever since every receiver obviously experiences a timeout here. Since the counter values are continuously exchanged via the content of the Alive messages, eventually all processes reach agreement upon all counters that have stopped increasing. Hence, locally electing the process with minimal counter indeed leads to a correct implementation of $\Omega$.

## References

[**ADGFT04** ] Aguilera, M.K., Delporte-Gallet, C., Fauconnier, H., Toueg, S.: Communication-efficient leader election and consensus with limited link synchrony. In: Proc. PODC 04, ACM Press (2004) 328–337

[**MOZ05** ] Malkhi, D., Oprea, F., Zhou, L.: $\Omega$ meets paxos: Leader election and stability without eventual timely links. In: Proc. DISC 05, Springer-Verlag (2005)

[**HMSZ05:TR** ] Hutle, M., Malkhi, D., Schmid, U., Zhou, L.: Chasing the weakest system model for implementing omega and consensus. Research Report 74/2005, Technische Universität Wien, Institut für Technische Informatik, Treitlstr. 1-3/182-2, 1040 Vienna, Austria (2005)