Lower Bounds for Asynchronous Consensus

Leslie Lamport
Microsoft Research

30 September 2002

Consensus is usually expressed in terms of agreement among a set of processes. Instead, we characterize it in terms of three classes of agents:

Proposers A proposer can propose values.

Acceptors The acceptors cooperate in some way to choose a single proposed value.

Learners A learner can learn what value has been chosen.

In the traditional statement, each process is a proposer, an acceptor, and a learner. However, in a distributed client/server system, we can also consider the clients to be the proposers and learners, and the servers to be the acceptors.

The consensus problem is characterized by the following three requirements, where N is the number of acceptors and F is the number of acceptors that must be allowed to fail without preventing progress.

Nontriviality Only a value proposed by a proposer can be learned.

Safety At most one value can be learned.

Liveness If a proposer p, a learner l, and a set of $N - F$ acceptors are non-faulty and can communicate with one another, and if p proposes a value, then l will eventually learn a value.

Nontriviality and safety must be maintained even if at most M of the acceptors are malicious, and even if proposers are malicious. By definition, a learner is non-malicious, so the conditions apply only to non-malicious learners. A malicious acceptor by definition has failed, so the $N - F$ acceptors in the liveness condition do not include malicious ones. Note that M is the maximum number of failures under which safety is preserved, while F is the maximum number of failures under which liveness is ensured. These parameters are, in principle, independent. Hitherto, the only cases considered have been $M = 0$ (non-Byzantine) and $M = F$ (Byzantine). If malicious failures are expected to be rare but not ignorable, we may assume $0 < M < F$. If safety is more important than liveness, we might assume $F < M$.

The classic Fischer, Lynch, Paterson result [4] implies that no purely asynchronous algorithm can solve consensus. However, we interpret "can communicate with one another" in the liveness requirement to include a synchrony requirement. Thus, nontriviality and safety must be maintained in any case; liveness is required only if the system eventually behaves synchronously. Dwork, Lynch, and Stockmeyer [3] showed the existence of an algorithm satisfying these requirements.

Here are approximate lower-bound results for an asynchronous consensus algorithm. Their precise statements and proofs will appear later.

Approximate Theorem 1 If there are at least two proposers, or one malicious proposer, then $N > 2F + M$.

Approximate Theorem 2 If there are at least two proposers, or one malicious proposer, then there is at least a 2-message delay between the proposal of a value and the learning of that value.

Approximate Theorem 3 (a) If there are at least two proposers whose proposals can be learned with a 2-message delay despite the failure of Q acceptors, or there is one such possibly malicious proposer that is not an acceptor, then $N > 2Q + F + 2M$.

(b) If there is a single possibly malicious proposer that is also an acceptor, and whose proposals can be learned with a 2-message delay despite the failure of Q acceptors, then $N > \max(2Q + F + 2M - 2,\ Q + F + 2M)$.

These results are approximate because there are special cases in which the bounds do not hold. For example, Approximate Theorem 1 does not hold in the case of three distinct processes: one process that is a proposer and an acceptor, one process that is an acceptor and a learner, and one process that is a proposer and a learner. In this case, there is an asynchronous consensus algorithm with $N = 2$, $F = 1$, and $M = 0$.

The first theorem is fairly obvious when $M = 0$ and has been proved in several settings. For $M = F$, it was proved in the original Byzantine agreement paper [6]. The generalization is not hard. Results quite similar to the second theorem have also been proved in several settings [2]. The third theorem appears to be new.

The bounds in these theorems are tight. Castro and Liskov [1] present an algorithm satisfying the bounds of the first theorem. A future paper will describe a new version of the Paxos algorithm [5] that obtains agreement in two message delays under the weakest conditions allowed by the third theorem.

References

[1] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, pages 173–186. ACM, 1999.

[2] Bernadette Charron-Bost and André Schiper. Uniform consensus is harder than consensus (extended abstract). Technical Report DSC/2000/028, École Polytechnique Fédérale de Lausanne, Switzerland, May 2000.

[3] Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM*, 35(2):288–323, April 1988.

[4] Michael J. Fischer, Nancy Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 32(2):374–382, April 1985.

[5] Leslie Lamport. The part-time parliament. *ACM Transactions on Computer Systems*, 16(2):133–169, May 1998.

[6] Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, April 1980.