Probabilistically-Safe Broadcast Algorithms

Pascal Felber
Institut EURECOM
2229 route des Crêtes, BP 193
06904 Sophia Antipolis, France

Fernando Pedone
Hewlett-Packard Laboratories
Software Technology Laboratory
Palo Alto, CA 94304, USA

1 Motivation

Message ordering abstractions, and more specifically group communication protocols, are very useful for the design of reliable distributed systems. Briefly speaking, message ordering abstractions ensure agreement on which messages are delivered in the system and on the order such messages are delivered. Many problems related to reliable and highly-available computation have been solved using one-to-many communication primitives with total-order guarantees.

Until recently, however, scalability has been the Achilles' heal of reliable one-to-many protocols. It has been demonstrated for example that group communication protocols do not scale well past a couple of hundreds of processes and degrade rapidly when executed across wide-area networks [BHO$^+$99].

Recent research has shown that algorithms providing probabilistic guarantees are a promising alternative for such environments. Provided that they are "adequately" high, probabilistic guarantees are sufficient for many applications. Several probabilistic protocols have been proposed to solve various group communication-related problems such as reliable broadcast and group membership. All the protocols we are aware of (e.g., [HB96, BHO$^+$99]) ensure deterministic safety. We propose a specification of probabilistic atomic broadcast with both probabilistic liveness and safety guarantees. We argue that probabilistic safety is a useful property if safety violations are very infrequent and processes can determine when they happen.

2 The PABCast Specification

We study the problem of probabilistic atomic broadcast and take into account not only probabilistic liveness but also probabilistic safety properties. We believe that many applications can take advantage of faster and more scalable algorithms without deterministic safety, if safety violations are infrequent and can be detected.

Consider a system composed of a finite set Π of processes that communicate by message passing. The probabilistic atomic broadcast problem—PABCast—is defined by the primitives broadcast(m) and deliver(m), which guarantee *Agreement*, *Order*, *Validity*, and *Integrity*. The former three properties are probabilistic and the latter is deterministic. In the following, p and q are two processes in Π.

Probabilistic Agreement. If p delivers m, then with probability γ_a, q also delivers m.

Probabilistic Order. If p and q both deliver m and m', then with probability γ_o they do so in the same order.

Probabilistic Validity. If p broadcasts message m, then with probability γ_v, p delivers m.

Integrity. Every message is delivered at most once at each process, and only if it was previously broadcast.

PABCast generalizes the traditional atomic broadcast properties [HT93] to allow messages to be delivered by any subset of the processes (from probabilistic agreement), out of order (from probabilistic order), and not at all (probabilistic validity).

Probabilistic agreement and order are independent of each other, as illustrated in Figures 1 and 2. In the run depicted in Figure 1, all processes deliver messages m and m', but p_1 and p_3 deliver m before m' and p_2 delivers m' before m, thus, agreement is satisfied but order is not. In Figure 2, p_2 does not deliver m', but all processes deliver m before m'', and p_1 and p_3 deliver m, m', and m'' in the same order, so order is satisfied but agreement is not.

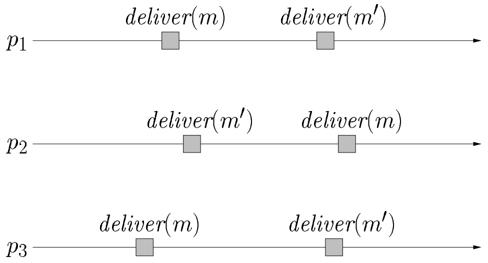

Figure 1: Run with agreement but no order

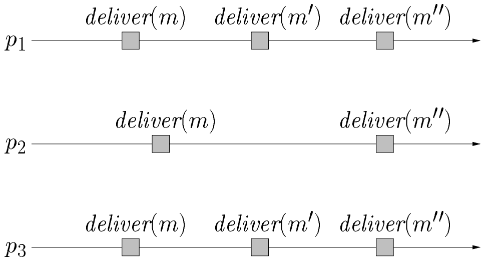

Figure 2: Run with order but no agreement

Protocols such as [HB96] provide probabilistic agreement and validity, but preserve deterministic order and integrity. The motivation is to permit provision of higher-level protocols that remain safe even when the system degrades from the model, due the "eventual convergence" property of the underlying protocol.

We do however believe that there is little to gain by keeping order deterministic when agreement is probabilistic. Indeed, for a given process, a missed message is generally as bad as a message delivered out-of-order. In both cases, the process must be able to take recovery actions to deal with such situations. Therefore, our PABCast specification allows order to be also probabilistic. The key requirement is that the process eventually learns about the missed or out-of-order message. When the probabilities of violating the protocol's properties are very small, the extra cost of dealing with such exceptional cases is amortized by the higher efficiency and scalability of the protocol.

3 Implementing PABCast

In [FP01], we present a protocol that implements the PABCast specification. This protocol is resilient to message losses and f process failures, where f is a parameter of the protocol. Processes execute a sequence of rounds, and during a round they can vote for broadcast messages. Among the protocol features, messages that receive $f+1$ votes in a round—a very frequent situation in practice—are delivered by all correct processes in the same order. It is therefore easy to distinguish a deterministically ordered message from a message that may not be correctly ordered.

We have analyzed the probabilistic behavior of our protocol under various conditions. Analytical and simulation results demonstrate that our protocol is highly reliable and scalable, and that the number of out-of-order messages is small in most scenarios.

4 Discussion

Probabilistic protocols are a promising approach to increasing the scalability of distributed systems. While offering weaker guarantees than deterministic protocols, they are still of practical interest if these guarantees can be quantified and shown to be small. We believe that by carefully balancing between probabilistic liveness and safety guarantees, we can build other lightweight probabilistic protocols of broad interest for highly-scalable distributed computing.

References

[BHO+99] K. P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky. Bimodal multicast. *ACM Transactions on Computer Systems*, 17(2):41–88, May 1999.

[FP01] P. Felber and F. Pedone. Probabilistic atomic broadcast. Technical report, Bell Labs, Lucent, December 2001. Also appears as Hewlett-Packard Technical Report HPL-2002-69, 2002.

[HB96] M. Hayden and K. Birman. Probabilistic broadcast. Technical Report TR96-1606, Cornell University, Computer Science, September 1996.

[HT93] V. Hadzilacos and S. Toueg. Fault-tolerant broadcasts and related problems. In *Distributed Systems*, chapter 5. Addison-Wesley, 2nd edition, 1993.