Part II

The Reliable Broadcast Communication Abstraction

This part of the book is devoted to the implementation of reliable broadcast abstractions on top of asynchronous message-passing systems prone to failures. Each of these abstractions is defined by a set of properties, and any algorithm (that claims to implement it) must satisfy these propertiers. This abstraction-oriented approach allows us to (a) know when these broadcast abstractions can be implemented and when they cannot, and (b) reason on the algorithms that use them, in a precise way. This part of the book is composed of three chapters:

- Chapter 2 defines the *reliable broadcast* communication abstraction, and presents algorithms implementing it in the presence of process crash failures (system model $CAMP_{n,t}[\emptyset]$). These algorithms differ in the abstraction level they implement, namely in the additional quality of service (basic, FIFO, and causal order) they provide.
- Chapter 3 extends the results of the previous chapter, namely, it considers that channels may lose messages. To this end, it introduces the notion of a fair channel and the notion of an unreliable channel.
- Chapter 4 considers the case where some processes (not known in advance) can commit Byzantine failures (model $BAMP_{n,t}[\emptyset]$), and presents algorithms suited to this model.

Let us remember that the model parameter t denotes the maximum number of processes that can be faulty (crash or Byzantine failures according to the failure model). While, in a crash failure model with reliable asynchronous channels, a reliable broadcast communication abstraction can be built for any value of t, this is no longer true in a crash failure model with fair asynchronous channels, and in a Byzantine failure model. Chapter 3 and Chapter 4 present corresponding computability bounds, and algorithms which are optimal with respect to these bounds.