



# ATOMIC BROADCAST ON VARIOUS MIDDLEWARES

**Niklaus Hirt**

**Diploma Thesis 1997/98**

**Professor: André Schiper**

**Supervisor: Rachid Guerraoui**

## Summary

---

### Subject

The subject of this diploma thesis is to develop and study possible implementations, providing **Atomic Broadcast** in JAVA. The solutions will use different **middlewares** and **algorithms**. Their respective **performance measures** will be compared with **theoretical estimations**, finally leading to conclusive results on reliability and speed.

### Solution

The implementation of the TO-Broadcast primitive is based on a **three layer model** where:

- The first layer implements communication using three different middlewares (**Sockets / RMI / CORBA**).
- The second layer implements the protocol using three algorithms (**Sequencer, Skeen and Chandra Toueg**).
- The third layer is represented by the application.

The measures are confirming the estimations, considering the percentual ratios. However, the exact values have only be approached as we found that the combination of middleware/algorithm is quite sensitive to the execution environment (e.g. network charge).

Sockets offer the fastest communication middleware. Then CORBA takes the second place, whereas RMI can be considered slow.

The algorithms also unveil a performance gradient, where Sequencer is only little faster than Skeen's algorithm, and Chandra and Toueg's algorithm is slow, due to its complexity.

Based on a proper registry server concept, a fully **transparent implementation** has been achieved, completely hiding the underneath structure from the user. Providing the possibility to easily equip existing software with Atomic Broadcast properties, by adding few method calls.

### Future Work

The provided work can be extended in several directions as there are:

- Implementation of failure detectors for consensus.
- Offer more flexible and powerful registration possibilities (e.g. connect and disconnect servers, introducing state transfers)
- Improve performance by using other, more recent algorithms and/or other middlewares (e.g.Orbix).