# The Bully Algorithm

The bully algorithm is an election algorithm used to elect a new group coordinator in the case where the previous group coordinator has crashed. This algorithm allows processes to crash during an election, although it assumes that message delivery between processes is reliable. The bully algorithm also assumes that each process knows which processes have higher identifiers, and that it can communicate with all such processes.

There are three types of messages in this algorithm: an **election** message is sent to announce an election, an **answer** message is sent in response to an election message and a **coordinator** message is sent to announce the identity of the elected process – the new coordinator in our case. A process begins an election when it notices, through the failure detection algorithm, that the coordinator has failed. Several processes may discover this concurrently. Hence in every server process *i*, a list with all server processes, each associated with it’s own identifier will be available, along with a variable *electedi* to determine the group coordinator.

The process that knows it has the highest identifier can elect itself as the coordinator simply by sending a coordinator message to all processes with lower identifiers. If a process with a lower identifier notices a failure, it can begin an election by sending an election message to those processes that have a higher identifier and awaiting a time *T* for an answer message in response. If none arrive, it is assumed that all higher-ordered processes have failed, considers itself the coordinator and sends a coordinator message to all processes with lower identifiers announcing this. Otherwise, if the process does receive answer messages back, it waits a further period *T’* for a coordinator message to arrive from the new coordinator. If none arrive, it is most likely due to the coordinator message being dropped and the process begins another election.

If a process *pi* receives a coordinator message, it sets its variable *elected­i* to the identifier of the coordinator contained within it and treats that process as the coordinator. If a process receives an election message, it sends back an answer message and begins another election, unless it has begun one already. When a process is restarted after a failure, it must begin an election. If it has the highest process identifier, then it will decide that it is the coordinator and announce this to the other processes, hence “bullying” out the current coordinator.