Approximate Consensus in Presence of Crash Failures

Consider an asynchronous system consisting of n processes. named $1, 2, 3, \cdots, n$, with x_i denoting the input of process i.

All processes can communicate with each other.

At most f processes may crash.

The following algorithm performed by each fault-free process i achieves approximate consensus provided thet $n \geq 2f + 1$. Specifically, after r becomes sufficiently large at all the fault-free processes, for fault-free processes i, j, we will have y_i apporoximately equal to y_j (more specifically, for any positive constant ϵ, after r exceeds a certain threshold at all the nodes, $|y_i - y_j| < \epsilon$).

1. Initialization:
   $y_i := x_i$
   $r := 1$

2. Send message (y_i, r) to all the processes including self.

3. Wait until $(n - f)$ messages of the form $(*, r)$ are received (including message from self).

4. Update y_i = average of the $n - f$ values in the above $n - f$ messages. Note that the value is the first field in the tuple in each message.

5. $r := r + 1$

6. Go to step 2

The algorithm above is said to perform "asynchronous rounds", with r being the round index. Unlike a synchronous round, an asynchronous round may complete at different processes at very different times. However, due to the inclusion of round index in the messages, each process can use the appropriate messages in its computation in each round.

The above algorithm may be terminated after a large enough number of rounds (termination not shown in the code above) by when the y_i values for different nodes are guaranteed to be within a desired constant.