**Implementation of**  **Counter CRDT on Android**

**Counter CRDT alogrithm**



**Quasi-dynamic member**

In this project, it is assumed that all processes know its maximum peers in the future when it startup. Maybe some peer not start up yet, and some peer process is unaccessible.

**How to simulation partition?---Simple solution**

Normally, CRDT can support eventual consistency when partioned network heal up. The best design is that every process send its own state update message to peers perodically. So every process which re-accesss network can receive the update message, and get the latest state. For simplicty, I choose another way: when the process re-access network, it ask the peers for their latest states. I also simulate the process’ “detach” from the network with the death of the process, and simulate the process’ “reunite” with the newtork with restart of the process.

**Socket timeout—deal with “disconneced” process**

In my design, add timeout to socket operation, when timeout occurs, it means that the peer process is disconnected.

**HashMap store vector**

In the orignal algorithm, arrays are used to store vectors. However, for better compatibility, HashMap is used to store the vetors, and the key is the process id.