RPC failure semantic - at-least-once

What does a failure look like to the client RPC library?

- Client never sees a response from the server
- Client does not know whether the server processed the request

Simplest scheme - **at-least-once behavior**

- RPC library waits for response for time T, if none arrives, re-send the request
- Possibly repeat this a few times
- If still no response then return an error to the application

RPC failure semantic - at-most-once

- Problem with at-least-once behavior
  What if the request is "deduct $100 from bank account" ?

➡ At-least-once works well with idempotent requests

Another (better) RPC behavior - **at-most-once**

➡ Having Server RPC code detects duplicate requests returns previous reply instead of re-running handler

- How to detect a duplicate request?
  - Client includes unique ID (XID) with each request, and uses the same XID for re-send
  - Server checks an incoming XID in a table, if an entry is found, directly returns the reply