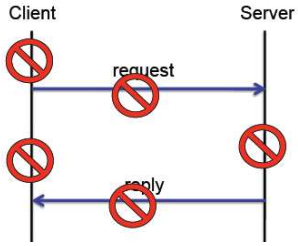
8. Fault Tolerance

Fejldefinitioner

- **Failure:** En fejl! Forskellige typer; crash, omission, timing, response, arbitrary.
- **Error:** Årsag til failure.
- **Fault:** Årsag til error. **Transient** (), **intermittent** (), **permanent** ().
- **Defect:** Årsag til fault.

Forventninger til DS



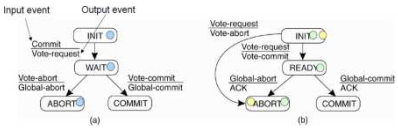
- **Availability:** Systemet er oppe og kører som forventet. (oppetid)
- **Reliability:** Om hvor længe systemet er oppe, før det går ned.
- **Safety:** Hvis noget går galt, er konsekvenserne ikke katastrofale.
- **Maintainability:** Systemet er let at fikse efter en fejl.

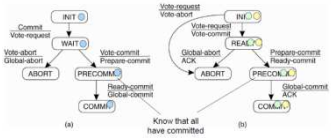
Klient-server kommunikation

1. **Klient kan ikke kontakte server:** Stubs skal synkroniseres/opdateres. Kan klares med exceptions…
2. **Request-besked forsvundet:** Bruge timers til at tjekke på ACK. Hvis ikke ACK modtages, så send ny besked.
3. **Server Crashes:** Kan crashe flere steder. Tre metoder for stub:
    - **At least once semantics:** Venter til server genstarter (eller rebind til ny). Bliv ved med at sende til vi får svar. RPC gennemføres mindst én gang.
    - **At most once semantics:** Giver op med det samme, og rapporterer fejl. RPC udføres max en gang.

    Det vi vil have, er dog **excactly once semantics**; umuligt.
4. **Reply-besked forsvundet:** Tilføje sekvensnumre på requests. Idempotent (get) – kan udføres mange gange. Mutator er mere kritiske.
5. **Klient crasher:** Resulterer i **orphan;** beregning i gang uden at kunne give resultat til nogen. Spilder CPU. Låser filer. Hvis klient rebooter, sender request på ny, og modtager tidligere resultat med det samme →forvirring. 4 løsninger:
    - **Orphan extermination:** Log alle RPCs. Tjek denne ved reboot og fjern orphans.
    - **Reincarnation:** Del tid op i nummererede epochs. Ved reboot→ start på ny epoch (orphans fra tidligere slåes ihjel).
    - **Gentle reincarnation:** Hvis orphans ejere ikke kan findes, slåes de ihjel.
    - **Expiration:** RPC får en tid T. Får løbende tildelt mere. Hvis klient venter tid T med at genstarte er alle orphans væk.

Distributed commit





- Alle eller ingen medlemmer i en gruppe skal udføre en operation.
- **2PC:** Hvis koordinator crasher, når alle processer er i READY er systemet låst, indtil den kommer op igen. **[TRANSPARENT]**
- **3PC:** Ingen process kan crashe i COMMIT, mens andre er i READY. Hvis bare én process er i PRECOMMIT, kan alle andre gå derhen. **[TRANSPARENT]**

Recovery

- **Forward:** Komme til fremtidig tilstand, hvor der ikke er fejl (skal kende fejltyperne)
- **Backward:** Lave checkpoints
- **Recovery line:**
- **Checkpointing (med 2PC)**
    - **Message logging**