Paxos commit

For dem som ønsker litt mer innsikt i detaljene i Paxos commit. Ikke pensum.

Paxos commit-algoritmen

Algoritmen slik den er gjengitt her, er hentet fra Gray og Lamports artikkel. Den er imidlertid noe forenklet/forkortet i forhold til hvordan en fullverdig versjon må se ut; dette er kommentert på slutten av algoritmen.

Ressurshåndtererne

Anta at det er K ressurshåndterere. Hver ressurshåndterer tildeles en id i form av et tall r mellom 1 og K.

La r være en ressurshåndterer.

Variable:

- $state_r \in$ {"working", "prepared", "committed", "aborted"}.
  - Initielt er $state_r$ = "working".

Aksjoner:

- $state_r$ = "working" og r finner ut at den er klar til å committe sin deltransaksjon:
  - $state_r$:= "prepared"
  - Send meldingen BeginCommit(r) til lederen.
  - Initiér en instans av Paxos consensus ved å sende meldingen Phase2a(r, 0, "prepared") til alle akseptorene.

- $state_r$ = "working" og r finner ut at den må abortere sin deltransaksjon:
  - $state_r$:= "aborted"
  - Rull tilbake deltransaksjonen.
  - (Valgfritt:) Send meldingen Phase2a(r, 0, "aborted") til alle akseptorene.
  - I tillegg bør nok r "vekke" opp lederen; det kan gjøres ved å sende meldingen BeginCommit(r) til lederen selv om r ikke selv kan committe.

- r mottar meldingen Prepare: Hvis $state_r \neq$ "working", så ignorer meldingen.
  Hvis $state_r$ = "working": Avgjør om deltransaksjonen kan committes eller må aborteres.
  Hvis kan committes:
  - $state_r$:= "prepared"
  - Send meldingen Phase2a(r, 0, "prepared") til alle akseptorene.

  Hvis må aborteres:
  - $state_r$:= "aborted"
  - Rull tilbake deltransaksjonen.
  - (Valgfritt:) Send meldingen Phase2a(r, 0, "aborted") til alle akseptorene.

- r mottar meldingen Commit: Hvis $state_r$ = "committed", så ignorer meldingen. Hvis $state_r \neq$ "committed" (da er alltid $state_r$ = "prepared"):
  - $state_r$:= "committed"
  - Commit deltransaksjonen.

- r mottar meldingen Abort: Hvis $state_r$ = "aborted", så ignorer meldingen. Hvis $state_r \neq$ "aborted" (da er alltid $state_r$ = "prepared" eller $state_r$ = "working"):
  - $state_r$:= "aborted"
  - Rull tilbake deltransaksjonen.

Akseptorene

Anta at det er N akseptorer. Hver tildeles en id i form av et tall mellom 1 og N.

La a være en akseptor.

Variable:

- For hver ressurshåndterer r: $prevVote_a[r] = (b,v)$.
  - Initielt er $prevVote_a[r] = (-1, \text{"none"})$ for alle r.
  - Når $prevVote_a[r] = (b,v) \neq (-1, \text{"none"})$, betyr det at den siste fase 2a-meldingen som a besvarte og som gjaldt r, var Phase2a(r, b, v). Da er $v \in \{\text{"prepared"}, \text{"aborted"}\}$.

- For hver ressurshåndterer r: $nextBallot_a[r] = b$, der b er det høyeste rundenummeret som a har sett som gjaldt r.
  - Initielt er $nextBallot_a[r] = 0$ for alle r.

Aksjoner:

- a mottar meldingen Phase1a(r, b) der $b > nextBallot_a[r]$:
  - $nextBallot_a[r] := b$
  - Send meldingen Phase1b(r, b, b', w, a) til lederen, der $(b', w) = prevVote_a[r]$.

  (Hvis $b \leq nextBallot_a[r]$, så ignorer meldingen.)

- a mottar meldingen Phase2a(r, b, v) der $b \geq nextBallot_a[r]$:
  - $nextBallot_a[r] := b$
  - $prevVote_a[r] := (b, v)$
  - Send meldingen Phase2b(r, b, v, a) til lederen.

  (Hvis $b < nextBallot_a[r]$, så ignorer meldingen.)

Lederen

Vi forutsetter at lederprosessen alltid er på en av de nodene som har en akseptorprosess. Lederen tildeles en id som er lik den id'en den tilhørende akseptorprosessen har.

La s være id'en til lederen.

Variable:

- For hver ressurshåndterer r: $lastTried_s[r] = b$.
  - Initielt er $lastTried_s[r] = 0$.
  - Når $lastTried_s[r] = b \neq 0$, betyr det at den siste avstemningsrunden som lederen initierte i forsøket på å få vite om r kan committe eller må abortere, var b. Vi partisjonerer rundenumrene mellom de potensielle lederne slik at leder s benytter rundenumrene s, s + N, s + 2N,... osv.

Aksjoner:

- Lederen mottar en BeginPrepare(r)-melding (og har ikke mottatt noen BeginPrepare-melding for r før dette):
  - Send meldingen Prepare til alle ressurshåndtererne unntatt r.

- Lederen har for en gitt r ikke fått klarhet i om r kan committe eller må abortere innen en viss tidsfrist. Lederen tar da initiativet til en ny avstemningsrunde:
  - $lastTried_s[r]$:= if b = 0 then s else $lastTried_s[r] + N$
  - Send meldingen Phase1a(r, b) til alle akseptorene, der $b = lastTried_s[r]$.

- Lederen har for en gitt r mottatt Phase1b(r, b, b', w, a)-meldinger med $b = lastTried_s[r]$ fra en majoritet av akseptorene:
  - Hvis samtlige slike meldinger har b' = -1 (da er alle w = "none"), så sett v lik "aborted". Ellers: Finn den meldingen som har størst b', og sett v lik den tilsvarende w-en (denne w-en vil alltid være ≠ "none").
  - Send meldingen Phase2a(r, b, v) til alle akseptorene.

- Lederen har for hver eneste r mottatt Phase2b(r, b, "prepared", a)-meldinger med $b = lastTried_s[r]$ fra en majoritet av akseptorene:
  - Send meldingen Commit til alle ressurshåndtererne.

- Lederen har for minst én r mottatt Phase2b(r, b, "aborted", a)-meldinger med $b = lastTried_s[r]$ fra en majoritet av akseptorene:
  - Send meldingen Abort til alle ressurshåndtererne.

Kommentarer

- Algoritmen inneholder ikke noen fase 3 for Paxos consensus-instansene; disse er erstattet av at leder sender én melding (Commit eller Abort) til samtlige ressurshåndterere når utfallet er klart. Mer i tråd med den grunnleggende ideen ville vært at lederen sender fase 3-meldinger til samtlige ressurshåndterere etterhvert som resultatet i hver enkelt instans foreligger. Ressurshåndtererne vet da det endelige resultatet senest når de har mottatt fase 3-meldinger for samtlige instanser. (I tillegg kan jo leder gjerne sende konklusjonen i form av en Commit- eller Abort-melding.)

- Algoritmen over sier ikke noe om hvordan en ressurshåndterer som av en eller annen grunn ikke får det endelige resultatet, skal få vite dette. Forenklet skjer det ved noen tilleggsmeldinger der ressurshåndtereren (etter en timeout) spør alle akseptorene hva resultatet var. Minst en av dem har en lederprosess som tar ansvaret for å formidle status til ressurshåndtereren.

- Vi har ikke tatt med hvordan valg av ny leder kan gjøres. Paxos commit tåler at flere noder starter opp lederprosesser. Vi kan for enkelhets skyld tenke oss at en akseptor som etter en viss tidsperiode ikke hører noe fra lederen, oppretter en lederprosess på sin egen node. Den egentlige algoritmen for ledervalg bør gjøre noe mer avansert enn dette.

Oppgaver

Anta at det er 5 noder som initielt har følgende prosesser:
Node 1: En ressurshåndterer, en akseptor og lederen
Node 2: En ressurshåndterer og en akseptor
Node 3: En ressurshåndterer og en akseptor
Node 4: En ressurshåndterer
Node 5: En ressurshåndterer

1. Forklar forløpet i Paxos commit hvis alle ressurshåndtererne kan committe sin deltransaksjon og hver av Paxos consensus-instansene bare trenger én avstemningsrunde.

2. Forklar forløpet hvis ressurshåndtererne på node 1-4 kan committe, mens node 5 må abortere sin deltransaksjon, og hver av Paxos consensus-instansene bare trenger én avstemningsrunde.

3. Forklar forløpet hvis alle ressurshåndtererne kan committe sin deltransaksjon, men nettverket til node 5 er veldig tregt, så det drøyer med å komme noen respons fra ressurshåndtereren på node 5.

4. Forklar forløpet hvis alle ressurshåndtererne kan committe sin deltransaksjon og alle consensus-instansene har påbegynt avstemningsrunde 0, men deretter blir nettverket til node 3 veldig tregt slik at det drøyer med å komme respons fra akseptorprosessen på denne noden.

Anta at vi er i en situasjon der ressurshåndtereren på node 5 har fremmet dekretet "prepared" og lederen (på node 1) har fremmet dekretet "aborted" i samme instans av Paxos consensus (altså den instansen som ble initiert av ressurshåndtereren på node 5).

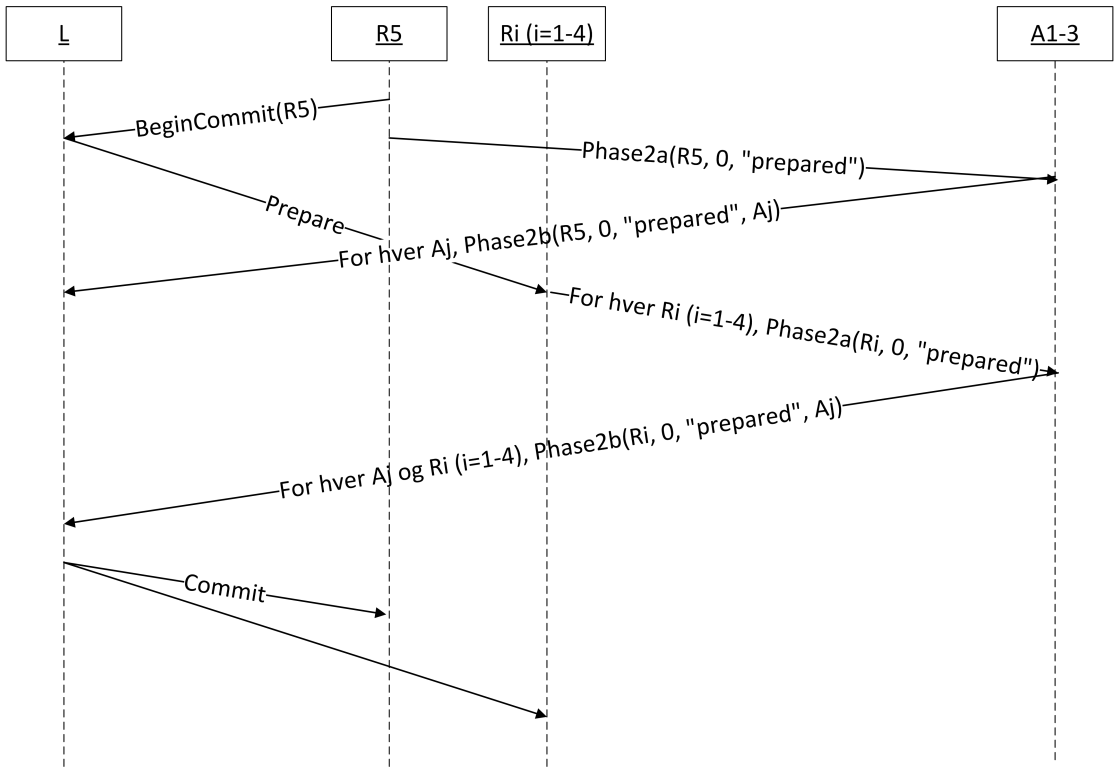
5. Gi et eksempel på et forløp der dette kan skje.

6. Gi et eksempel på et videre forløp der det i instansen blir fullført en avstemningsrunde med dekretet "aborted".

7. Gi et eksempel på et videre forløp der det i instansen blir fullført en avstemningsrunde med dekretet "prepared".

Løsningsforslag

Oppgave 1.



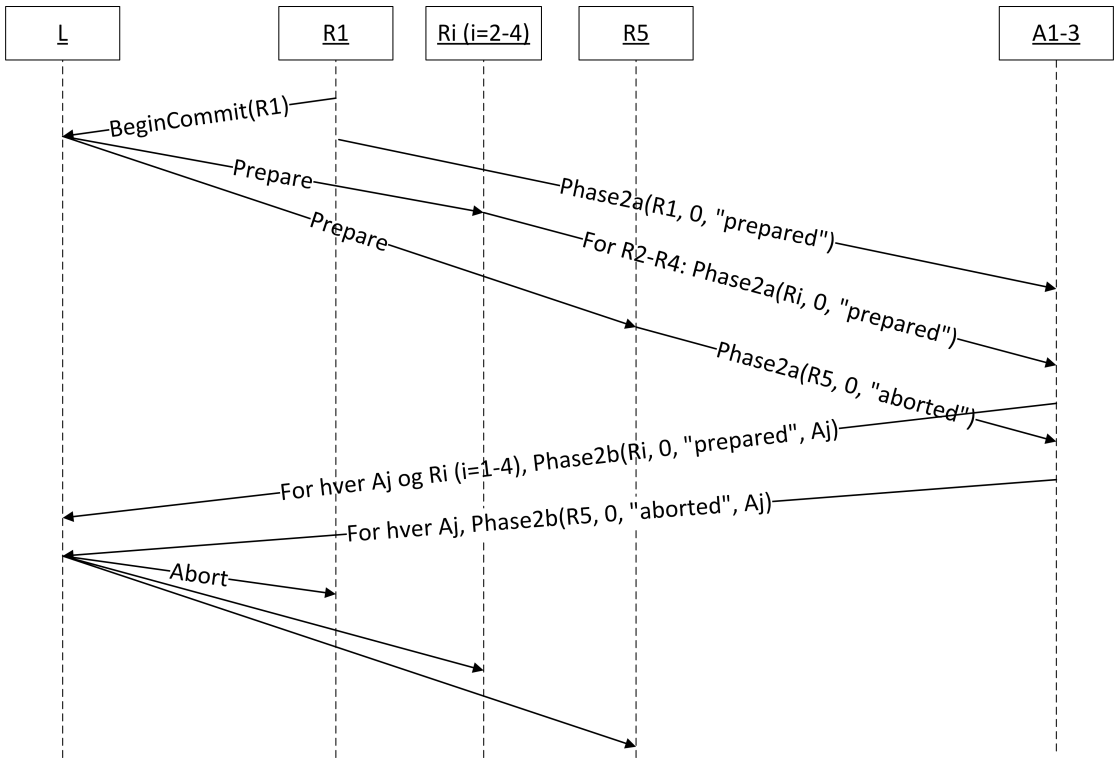
- L er lederprosessen. Ri er ressurshåndterer nr. i, der i=1,2,3,4,5. Aj er akseptorprosess nr. j, der j=1,2,3. Her har vi antatt at det er R5 som blir først ferdig med sin deltransaksjon. Figuren viser R1 til R4 som én felles "stolpe", og det samme for A1-3, for å forenkle fremstillingen litt.

- Når R5 er klar til å committe (og ikke har hørt noe fra de andre ennå), sender den BeginCommit til L og starter sin instans av Paxos consensus ved å sende Phase2a-meldinger med rundenummer 0 og dekretet "prepared" til alle akseptorene. Akseptorene besvarer med Phase2b-meldinger til L.

- Når L mottar BeginCommit-meldingen fra R5, sender den en Prepare-melding til hver av de andre Ri-ene. Når de andre Ri-ene mottar denne meldingen, gjør de på samme måte som R5: Hver starter sin instans med en Phase2a-melding med rundenummer 0 og dekretet "prepared", og akseptorene besvarer disse direkte til lederen.

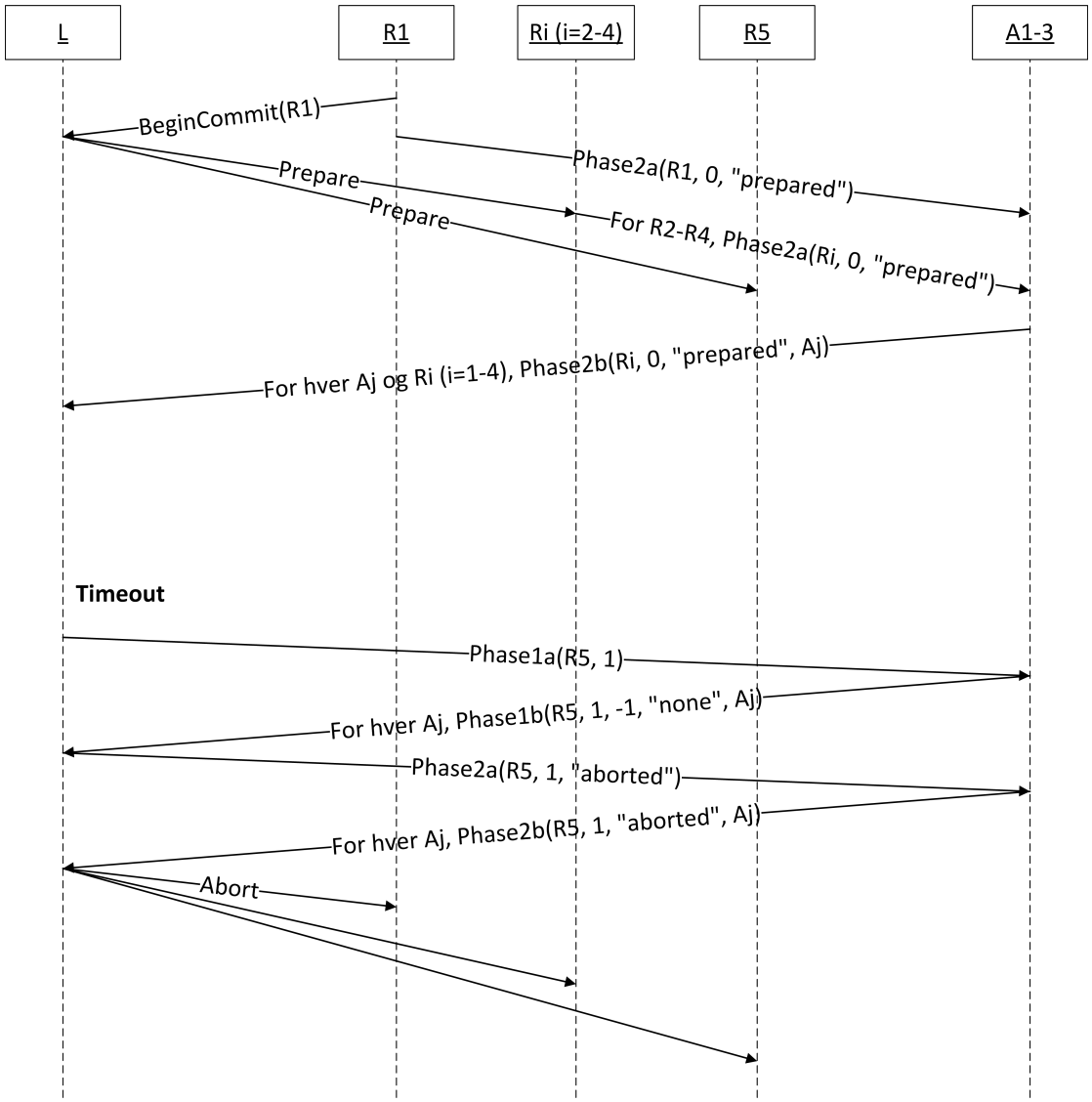
- Når L har mottatt Phase2b-meldinger med dekretet "prepared" fra et flertall av akseptorene for hver Paxos consensus-instans, har den oversikt over det endelige resultatet: Samtlige instanser hadde dekretet "prepared", så lederen sender Commit til alle Ri-ene.

Oppgave 2.



- Her har vi antatt at R1 er den som først blir ferdig med sin deltransaksjon og varsler L. Deretter sender L en Prepare-melding til de andre Ri-ene. Alle Ri-ene melder inn resultatet av sin deltransaksjon i form av Phase2a-meldinger til akseptorene: R1-4 med dekretet "prepared", R5 med dekretet "aborted". Hver akseptor besvarer med Phase2b-meldinger til L. L kan sende en Abort-melding til alle Ri-ene i det øyeblikket den har fått inn Phase2b-meldinger med dekretet "aborted" fra en majoritet av akseptorene. (Faktisk kan L sende en Abort-melding straks den mottar den første av Phase2b(R5, 0, "aborted", Aj)-meldingene. Dette gjelder imidlertid bare for runder med rundenummer 0.)

Oppgave 3.

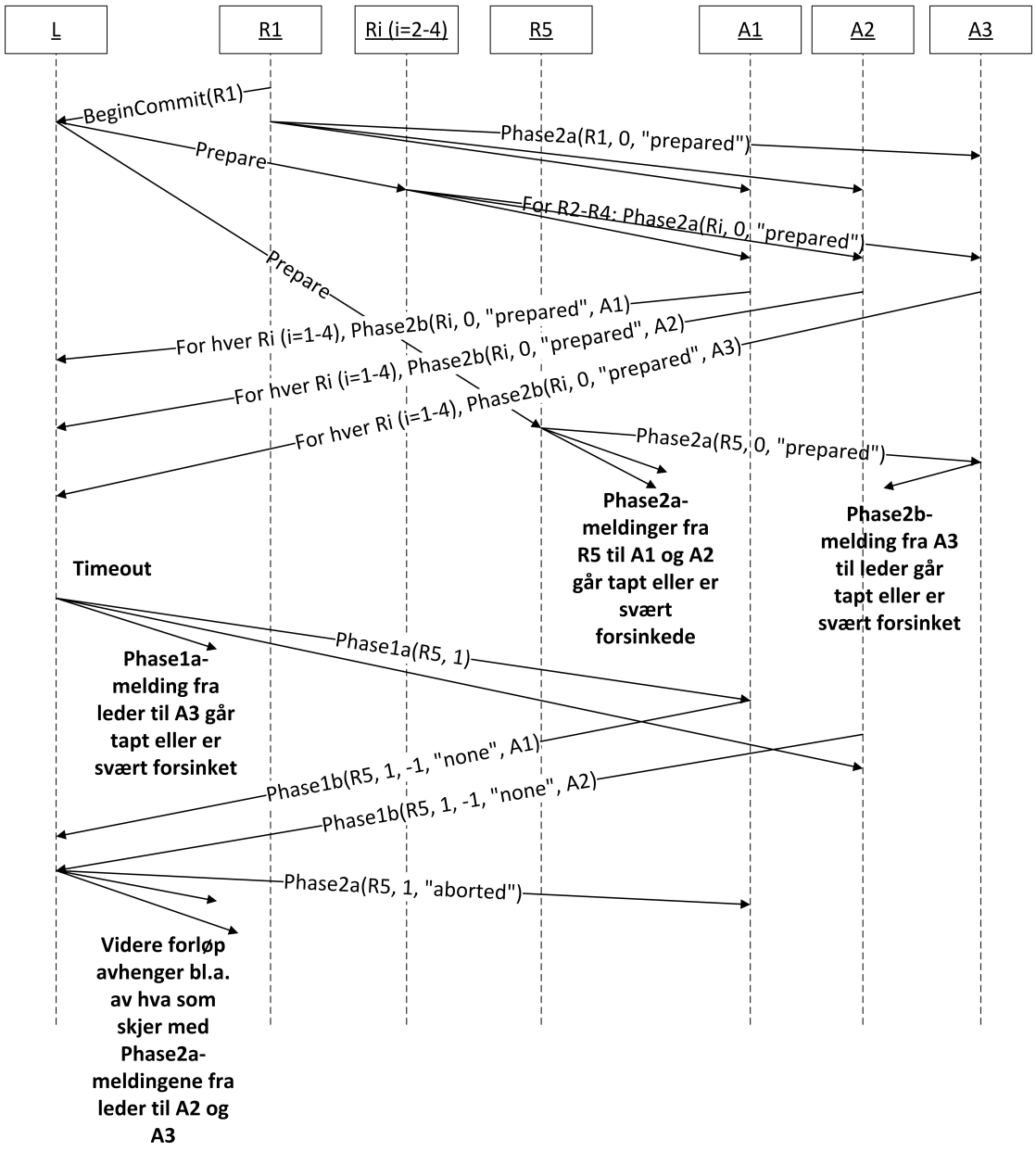


- L sender Prepare til alle, men R5 er treg med å svare (eller nettet til den er nede). Aj-ene sender Phase2b-meldinger for de Phase2a-meldingene de mottar, men det kommer aldri noen Phase2a-meldinger til dem fra R5.
- Etter en timeout påstarter L en ny avstemningsrunde for de instansene der Phase2b-meldinger ikke foreligger; i dette tilfellet gjelder det R5. Nye runder startes som vanlig med Phase1a-meldinger. Når L har mottatt svar fra Aj-ene (eller en majoritet av dem) og får vite at intet dekret er kjent, sender L en Phase2a-melding med dekretet "aborted", som så blir vedtatt.

Oppgave 4.

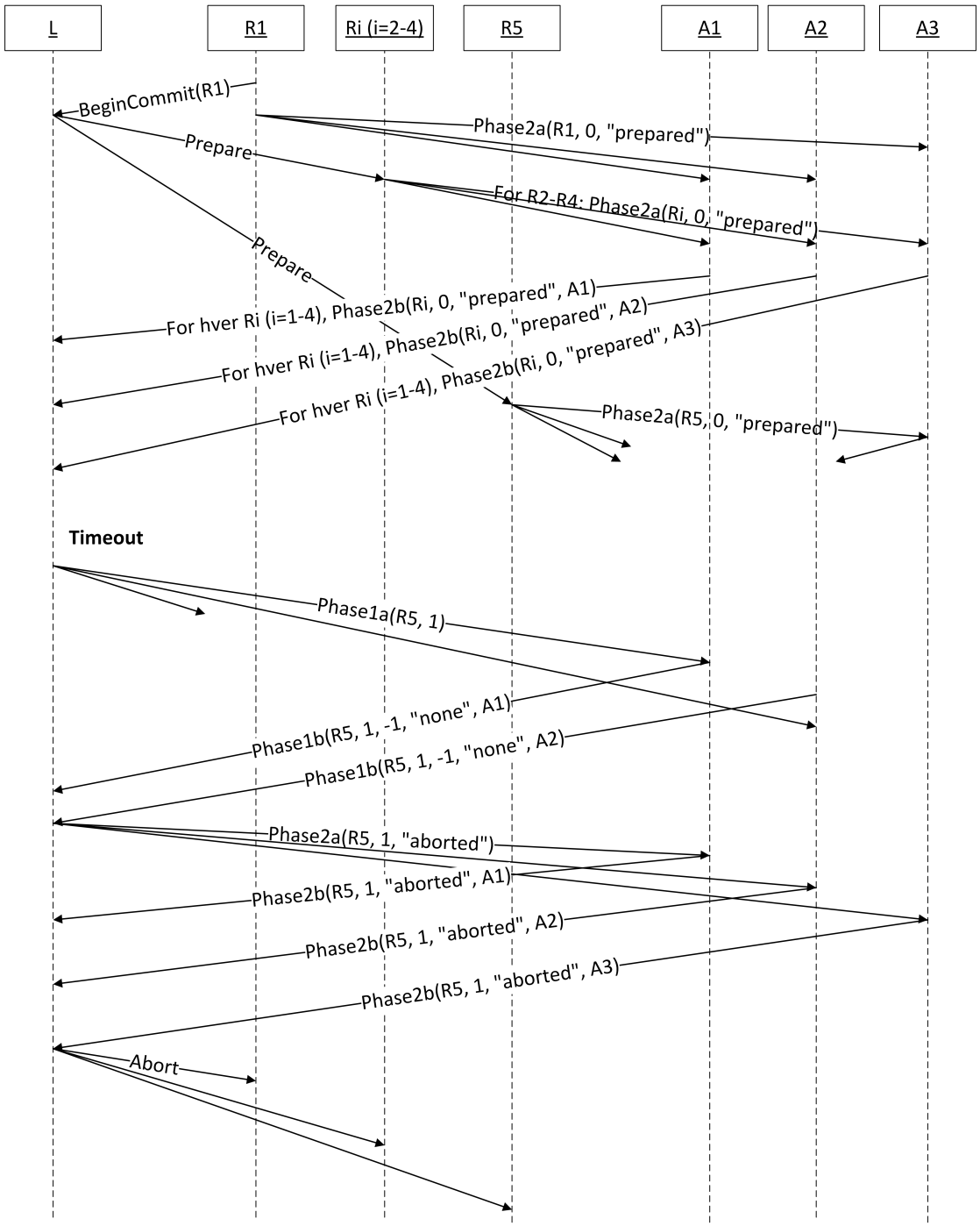
Her vil forløpet være omtrent som i oppgave 1, med unntak av A3 som ikke svarer. Det spiller ingen rolle, for så lenge en majoritet av akseptorene svarer, vil protokollen ha progresjon.

Oppgave 5.



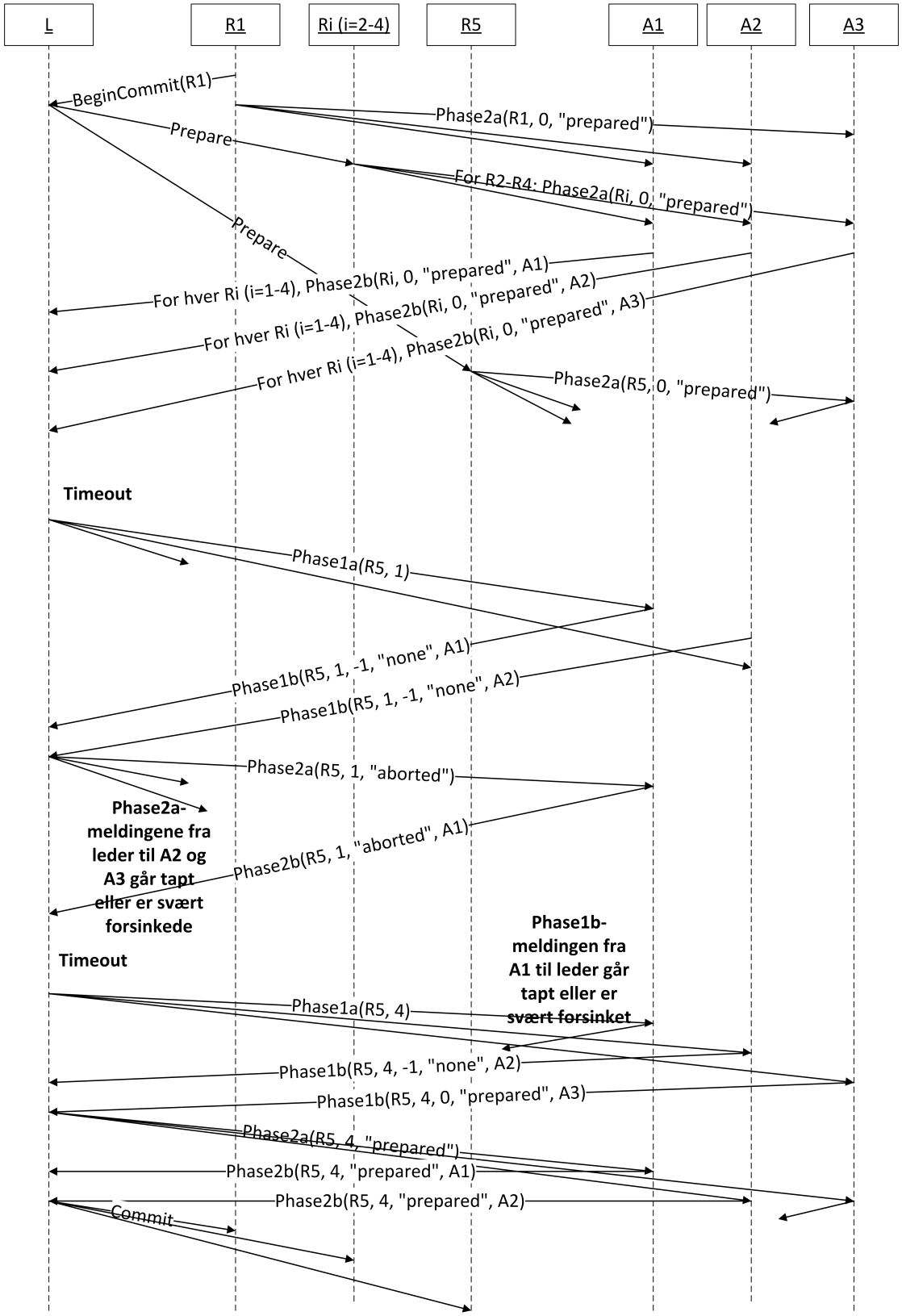
- Protokollen starter som vanlig, men i eksempelet over skjer det deretter noe med nettet mellom R5 og A1-A2 slik at det drøyer før Phase2a-meldingene kommer frem (eller meldingene går tapt). A3 mottar Phase2a-meldingen fra R5, men dens Phase2b-svar til L går tapt.
- I mellomtiden har L timet ut og starter en ny avstemningsrunde for R5 ved å sende Phase1a-meldinger til alle akseptorene. A1 og A2 får disse meldingene, men ikke A3. A1 og A2 returnerer Phase1b-meldinger til L.
- A1 og A2 utgjør en majoritet, så L kan fortsette. Siden A1 og A2 ikke visste om noe forslag til dekret, foreslår L dekretet "aborted" og sender dette til alle akseptorene. A1 mottar denne Phase2a-meldingen. Så nå foreligger to forslag til dekret for R5: "prepared" fra R5 selv (A3 vet om dekretet "prepared"), og "aborted" fra L (A1 vet om dekretet "aborted").

Oppgave 6.



- Her mottar og besvarer alle (eller en majoritet av) akseptorene Phase2a-meldingen fra L.
- Når L mottar svarene på den nye avstemningsrunden, vet den at dekretet for R5 er "aborted", så L sender Abort til samtlige ressurshåndterere.

Oppgave 7.


L
R1
Ri (i=2-4)
R5
A1
A2
A3
BeginCommit(R1)
Phase2a(R1, 0, "prepared")
Prepare
For R2-R4: Phase2a(Ri, 0, "prepared")
Prepare
For hver Ri (i=1-4), Phase2b(Ri, 0, "prepared", A1)
For hver Ri (i=1-4), Phase2b(Ri, 0, "prepared", A2)
For hver Ri (i=1-4), Phase2b(Ri, 0, "prepared", A3)
Phase2a(R5, 0, "prepared")
Timeout
Phase1a(R5, 1)
Phase1b(R5, 1, -1, "none", A1)
Phase1b(R5, 1, -1, "none", A2)
Phase2a(R5, 1, "aborted")
Phase2a-meldingene fra leder til A2 og A3 går tapt eller er svært forsinkede
Phase2b(R5, 1, "aborted", A1)
Timeout
Phase1b-meldingen fra A1 til leder går tapt eller er svært forsinket
Phase1a(R5, 4)
Phase1b(R5, 4, -1, "none", A2)
Phase1b(R5, 4, 0, "prepared", A3)
Phase2a(R5, 4, "prepared")
Phase2b(R5, 4, "prepared", A1)
Phase2b(R5, 4, "prepared", A2)
Commit

- I dette tilfellet er det bare A1 som mottar Phase2a-meldingen med dekretet "aborted" fra L. Etter en ny timeout prøver L nok en avstemningsrunde (denne gangen med rundenummer 4, siden dette er neste rundenummer som er reservert for node 1). A2 og A3 svarer på denne Phase1a-meldingen. A2 har ikke sett noe dekret hittil, mens A3 har sett dekretet "prepared". Når L får Phase1b-meldingene fra A2 og A3, vil L fremme dekretet "prepared" i Phase2a. Denne meldingen blir besvart av en majoritet, og dermed har alle Paxos consensus-instansene besvart med "prepared", og L kan meddele samtlige ressurshåndterere om at de kan committe.
- Eksempelet er ikke helt korrekt: Vi har forutsatt at A1 er på samme node som L. Det betyr at A1 umulig kan gå glipp av Phase1a(R5, 4)-meldingen fra L, og også at L alltid vil få svar fra A1. Eneste tilfelle der L vil fremme dekretet "prepared" i fase 2, er når minst en av aktuatorene i fase 1 rapporterer dekretet "prepared" og ingen av dem rapporterer det nyere dekretet "aborted". Siden lederen er den eneste som får initiere nye avstemningsrunder og derfor alltid vil motta en Phase1b(R5, .., .., "aborted", A1)-melding fra sin lokale akseptor A1, vil "prepared" derfor bare kunne bli fremmet i fase 2 i ytterligere avstemningsrunder hvis en annen node overtar lederansvaret. Hvis noden som har lederprosessen, ikke har noen akseptorprosess, kan vi imidlertid få en situasjon som beskrevet i figuren.