Forelesning 5

* Consistency
  + Absolute
    - Knyttet til alderen av en pakke
  + Related
    - Knyttet til to stykker data i forhold til hverandre.
* Temporal consistency
* Consumeren leser verdiene fra en buffer
  + Dlere problemer
  + Vi trenger å vite om en verdi er ny eller ikke
  + Ingenting hinderer produceren fra å ikke oppdatere verdien i bufferen
  + Det andre er når brukeren leser må man kunne se om det kommer av at nettet sender det på nytt.
* Alderen av dataen er bounded av de to timerene
  + Timeren for consumeren og produsenten
  + Alternaticet ville ha vært at alle har en felles klokke (Og det er ikke lett)
  + Vi kjenner ikke til alderen, vi vet bare at den ike er eldre enn en viss verdi
  + Dette er tilstrekkelig for å «establishe» validity durationen
  + Men vi kan ikke si noe om samtidighet.
    - Det fungerer dårlig for å finne ut om to verdier er skaffet samtidig
  + Løsningen er å bruke den synkroniserte versionene
    - Man bruker den synkroniserte variablen
    - Man sørger for at bus-arbitrene sier når man kan sample
    - TImeren starter med sync-variabblen. Man kan si at verdien ikke er eldre enn en viss verdi
  + Om man ser på literaturen i mange tilfeller sier de at et sanntidsnettvert bør ha veldig lite jitter, fordi de bruker sendingen av en pakke som en synkroniseringsmekanisme.
  + Man kan også innføre begrensninger på scheduleren, slik at man ikke får sett noe av det første tilfellet, men det er litt mer komplekst.
  + Central polling
    - Vi har en garantert transaction time (Gitt ingen feil)
    - Schedulingen er ikke spesifisert
* Temporal behaviour
  + Persodiske operasjoner er mulig
    - Også med flere perioder
  + Ingen retranmisjon om man har feil
    - Bortsett fra meldinger med ack
    - Oversampling kan være nødvendig for å tolllerere mangelen av en verdi i tilfellet av en feil
      * I disse tilfellene bruker man den tidligere verdien (Eller ekstrapolerer). Man later som om man får noe likevel.
  + Det er mye lettere å schedule on-line enn «Optimalt»
* Schadulability av periodisk trafikk
  + Anta N\_p transaksjoner per EC
  + De første transaksjonene av den i-ende periodisiske flyten vil være den m’te transaksjonen, hvor m er gitt ved
  + 6 produsenter, hver lager 3 strømmer
  + Og de har en viss prioritet
  + FIP messages (?)
* Om man kjenner til parametrene i en strøm (alt er meldinger av samme lengde)
  + Da kan man regne ut antall transaksjoner man kan få inn i en elementary cycle. Og ut ifra det kan man finne worst.case response time
* Vi antar at vi har noe gitt for det periodiske(?)
* Temporal analysis
  + Request vet tid t
  + Vil bli signalisert til BA(Bus arbiter) senest ved
* FIP er interesant for å hådtere periodisk trafikk
* Grenser kan utledes for sporadisk trafikk
  + Men de er direkte knyttet til polling period