Forelesning 1 april

EtherCat

* Det er en kabel fra en slave til den neste
  + Når man sender er det en short-cut tilbake
  + Alt må gå gjennom shift-registrene, etc for å sendes
* Fordeler
  + Rakst (Men trengs dette)
* Drawbacks
  + Fullstendig ukompatibelt med eternett,
  + og det krever spessialisert hardware
  + Bergrensende topologi

## SERCOS III

* topology
  + Man kan lage redundanser (f.eks en ring)
  + Eller en linje
* Cycle
  + Opp til 4 master Data temegram
  + Opp til 4 AT Device telegram
  + Et antall IP-meldinger (max. Duration)
* Man har en sykel som er delt opp i flere deler
  + Vi kaller dem channels
  + Det er garantert at trafikk vil gå gjennom nettet.
* Cycle
  + Real time channel er der I hver sykel for:
    - …
* Oppsummering
  + 254 noder kan settes på samme segmetn
    - Ring eller linje-topologi
  + Rent periodisk
  + Såkalt «Garantert og deterministisk kommunikasjon»
  + Er laget for å koble sammen drives
    - => Slave-enhetene
* Fordeler
  + Man har mulighet for redundans
  + Hotplug => Man kan koble
  + Man kan ta en slave-node og koble det til vanlig eternett
    - En port må holdes åpen
* Ulemper
  + Det koster
  + Krever non-standard hardware
* Eternett IP med time-synchronization
  + Det er ingen begrensninger på antall noder
  + Ingen modifikasjoner til 802.3 (Standerd-versjonen)
    - Milig
  + Node to node kommunikasjon gjennom en producer-consumer model
  + Men utenom denne modellen er det ikke noe mer som er gjordt for å oppnå sanntidskrav.
  + Fordeler
    - Standard
  + Ulemper
    - ….
* Ether Powerlink
  + Man har en manager (MN)
  + Synkrone delen er laget av slots
    - Kontinuerlig: CN polles ved hver sykel
    - Multiplexed: En gitt CN polles hver nte sykel
    - Synkrone utvekslinger
      * Sekvenser ac PollRequests etterfulgt ac PollResponse
      * Response er sendt i broadcast.
  + 250 kbps
  + Passfree switch
    - Med en gang man kjener addressen begynner man å sende med en gang
    - Man går fra 5 til mindre enn 1 mikrosekiund
    - Dette er et triks for å motvirke
  + Det er en dårlig ide å bruke eternett på denenmåten i stedet for å sende requests og responses hele tiden
    - Behold start of cycle, men ikke bruk poll request og poll response hele tiden
    - Sqitchene ville ha fikset dette ganske fint
    - Man ville ha økt effiktiviteten en hel del.
    - (Den var fremdeles ganske suksessfult på markedet ….)
      * Ikke sett noe som ikke passer oppå eternett
  + Oppsummering
    - Tillater 240 noder på et segment
      * Topologi ifølge eternett
    - Fast-switches
      * Kan gi ulemper om man sender en pakke som var gal, og man sier den er korrekt
    - Fordeler
      * Noe med en konstant cycle (Om det ikke er feil)
      * Line topology er mulig
      * Man antar at det ikke er noen feil, eller noen timing-feil
      * Alle noder som ikke er compliant med eternett-polling går det fing
      * Det er et application layer
  + Ulemper
    - Har bare et begrenset subsett av de nødvendige konseptene
    - Lav effitivitet
    - Kan ikke jobe med vanlig eternett
    - Trenger spessielt designede switches og hubs
    - Veldig følsomt ovenfor feil
    - Tenker ikke på timing errors

## Nåværende eternett-contenders

EPA (Real time eternett epa)

* Topologi
  + Antall noder er kun begrenset av fysiske constraints
  + Alle noder må snakke
    - Ved hver elementary cycle

Andre

* Standarder
  + Profinet
  + AFDX
    - Mye brukt innenfor fly
    - Switched eternett, hvor man gjør trafic controll ved inngangen.
  + MODBUS på TCP/IP
    - Ingen sanntid, bare en protokoll på toppen ac TCP
  + Vnet/IP
* TT-Ethernet
  + Ikke standard (for øyeblikket)
  + Rent time-triggered
* I en switch
  + Temporal definitions
    - T\_mux : Multiplexing delay
    - T\_queue: queuing delat
    - T\_trans; Oacket transmission delat = t\_frame + t\_mux 0 t\_queue + t\_os
  + Switched eternett
    - La oss se på bar en output port
* Maksimalt minne som trengs for å sende noe i switchen er gitt ac den største avstanden mellom kurvene
  + Det sier ogås noe om hvordan man kan dimmensjonere en switch for å unngå buffer overflow.
* Switch backlog (FIFO)
  + B: Verdien ved det siste indlection-pointet
  + Backlog og Backlog estimation
    - B\_k repressenterer bmrsty-ness (Det som er glatt gir ikke store problemer)
      * Problemet ligger i bursty-nessen.
      * Man bør prøve å regulere den om man vil minimere mengden data som sendes
* Network calculus
  + Man unngå buffer overflod
  + Om man vil unngp delats bør man kontrollere b\_k
* Man kan bruke en token bucket capaciry
  + Fill rate r: 1 toke per 1/r sekunder
* Når en pakke med n bytes kommer fjerner man n tokens og sender det videre
* Backlog er per egress queue
* Analysen er gyldig for en switch (Kan utvides )