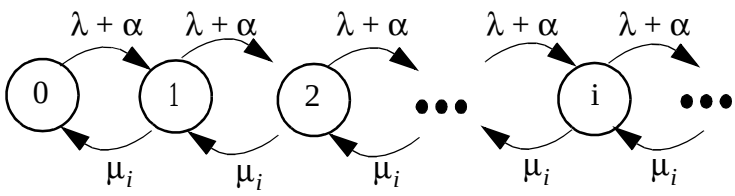
Oppgave 1 Ytelse i ett nettelement

a) *Beskriv systemet ved Kendalls notasjon og lag en Markov modell av NE_i hvorkun én pakke kan betjenes ad gangen og bufferet har uendelig kapasitet, $B = \infty$. Beregn avviklet trafikk og forventet antall i kø når $\lambda = 10$, $\alpha = 30$ og $\mu_i = 50$ [pakker/sekund].*

Kendall notasjon: M/M/1 - dvs. et enbetjener system med uendlig kø

Markov modell (tilstandsdiagram)



Figur 1.1 Tilstandsdiagram for køsystem med én betjener

Avviklet trafikk blir (formel (6.54) i kompendiet):

$$A' = \sum_{i=1}^{\infty} 1 \cdot p_i = 1 - p_0 = A = \frac{\lambda + \alpha}{\mu_i} = 0.8 \qquad (1.1)$$

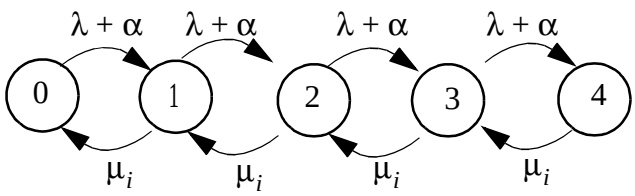
Ettersom det er uendlelig kø og derved intet pakketap så er tilbudt trafikk lik avviket trafikk, $A = A' = 0.8$.

Forventet antall i kø er (formel (6.56) i kombendiet som er formel (43) kombinert med (31) i formelsamling):

$$\mathrm{E}(Q) = \mathrm{E}(I) - A = \sum_{i=1}^{\infty} i p_i - A = \frac{A}{1-A} - A = \frac{A^2}{1-A} = \frac{0.8^2}{0.2} = 3.2 \qquad (1.2)$$

b) *Anta nå at bufferets kapasitet settes til $B = 3$. Tegn opp fullstendig tilstandsdiagram for dette systemet og sett opp ligninger som bestemmer tilstandssannsynlighetene i dette diagrammet. Løs ligningsettet med verdier $\lambda = 10$, $\alpha = 30$ og $\mu_i = 50$ [pakker/sekund]. Beregn avviklet trafikk og andel pakker avvist fra bruker X.*

Markov modell



Figur 1.2 Tilstandsdiagram for køsystem med én betjener

Kommentar: her er det mange som har kun modellert antallet i kø og har glemt betjeneren. De får da en tilstand mindre.

Tilstandsligninger

$$\begin{aligned} (\lambda+\alpha)p_0 &= \mu_i p_1 & p_1 &= ((\lambda+\alpha)/\mu_i)p_0 \\ (\lambda+\alpha)p_1 &= \mu_i p_2 & p_2 &= ((\lambda+\alpha)/\mu_i)^2 p_0 \\ (\lambda+\alpha)p_2 &= \mu_i p_3 \quad \Leftrightarrow & p_3 &= ((\lambda+\alpha)/\mu_i)^3 p_0 \\ (\lambda+\alpha)p_3 &= \mu_i p_4 & p_4 &= ((\lambda+\alpha)/\mu_i)^4 p_0 \end{aligned} \tag{1}$$

Normering gir verdien på p_0:

$$\sum_{i=0}^{4} p_i = (1 + (\lambda+\alpha)/\mu_i + ((\lambda+\alpha)/\mu_i)^2 + \ldots)p_0 = 1 \tag{2}$$

Innsatt $\lambda = 10$, $\alpha = 30$ og $\mu_i = 50$

$$\begin{aligned} p_0 &= 0.2975 \\ p_1 &= 0.2380 \\ p_2 &= 0.1904 \\ p_3 &= 0.1523 \\ p_4 &= 0.1218 \end{aligned} \tag{3}$$

Avviklet trafikk : $A' = 1 - p_0 = A(1 - p_4) = 0.7025$

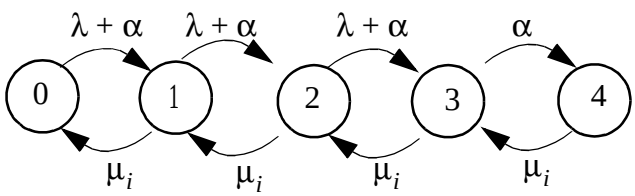
Avvist andel pakker fra bruker X (kan også settes opp direkte med henvisning til PASTA):

$$B = \frac{\lambda p_4}{\sum_{i=1}^{4} \lambda p_i} = p_4 = 0.1218 \tag{4}$$

Kommentar: Noen studenter har påpekt at formulieringen er noe uklar; derfor godtas både både svar med andel pakker avvist fra X i forhold til antall pakker fra X og i forholdt til totalt antall pakker.

c) *Anta at systemet er som i oppgave* b) *med unntak av at siste køplass er forbeholdt pakker som allerede er på ringen, dvs. som kommer fra oppstrømsnettelementet. Når vil pakker fra bruker X bli avvist da?*

Markov modell



Figur 1.3 Tilstandsdiagram for køsystem med én betjener

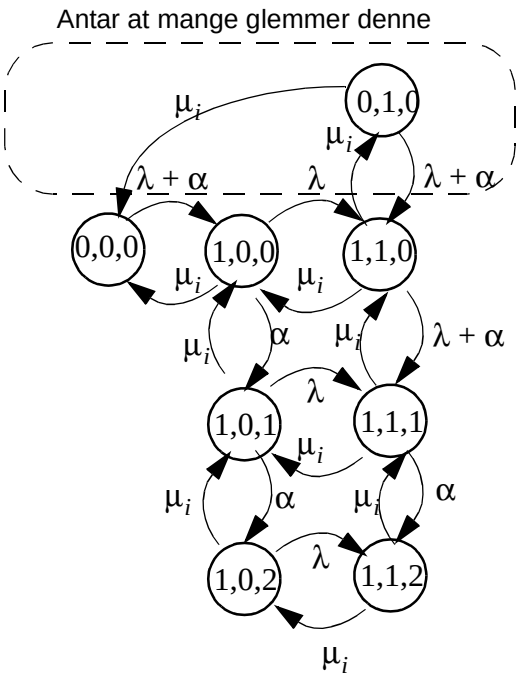
Dvs. eneste endring er overgangsintensiteten mellom tilstand 3 (2 i kø) og 4 (3 i kø) hvor ankomst av pakker fra bruker X ikke tillates å legges i kø. Altså: ankomst av pakker fra buker X avvises nå både i tilstand 3 og 4.

d) *Lag en ny Markov modell av hvor fortsatt kun én pakke kan betjenes ad gangen men kan nå betjenes av begge ringene. Spesifiser tydelig hvordan du har definert tilstandsvariablen(e) din(e). Betjeningsintensiteten er for hver av ringene. Bufferkapasiteten reduseres til $B = 2$, mens intensitetene er som før.*

Tilstandvariable i, j, k hvor

- i =primærring ledig (0) eller opptatt (1)
- j =sekundærring ledig (0) eller opptatt (1)
- k =antall i (primær)-kø (0,1,2)

Markov modell

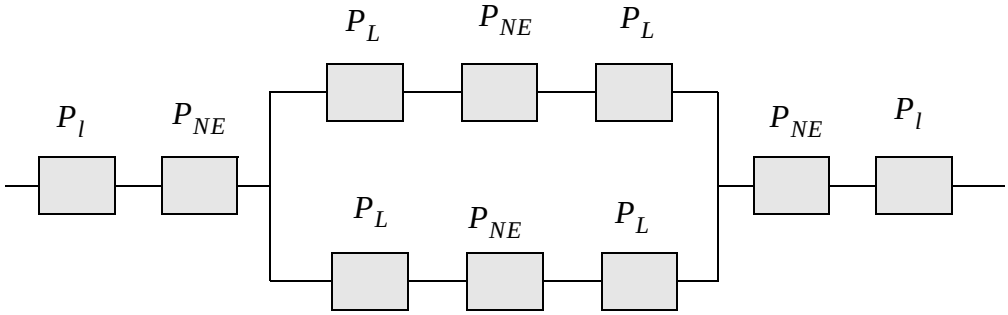


Figur 1.4 Tilstandsdiagram for køsystem med én betjener

(5)

Oppgave 2 Pålitelighet til tjenester og nettelement

a)



Tilgjengeligheten:

I dette tilfellet antas at alle nettelementer/linker/linjer feiler uavhengig av hverandre og at alle blir reparert uavhengig av hverandre (og da implisitt også av systemtilstand). P_X vil i dette tilfellet være den asymptotiske tilgjengeligheten for nettkomponent X, A_X.

Funksjonssannsynligheten:

I dette tilfellet antas at alle nettelementer/linker/linjer feiler uavhengig av hverandre. Det er <u>ingen</u> reparasjon av noen nettkomponent etter den har feilet. Dvs. vi kan kun beregne funksjonssansynligheten for et ikke-reparert system. P_X vil i dette tilfellet være den funksjonssannsynligheten for nettkomponent X, $R_X(t)$

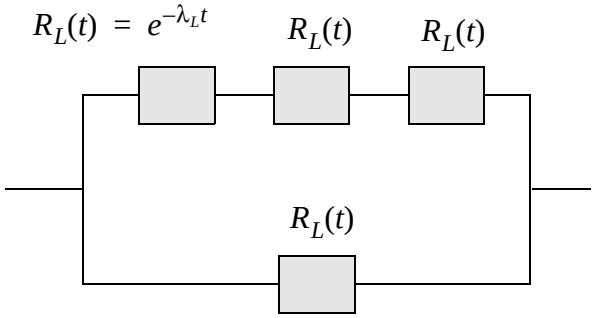
b) $$\boldsymbol{A}_{AC} = (A_l A_{NE})^2(1-(1-A_L^2 A_{NE})^2)$$

$$\mathbf{MTBF}_{AC} = \frac{\mathbf{MDT}_{AC}}{(1-\boldsymbol{A}_{AC})}$$

c) Det ventes at studentene ut fra kjennskap til Poissonprosessen, "rå hukommelse", formelsamling, m.v. presterer at påliteligheten ved konstant rate/intensitet er $R_X(t) = e^{-\lambda_X t}$. Da er nedenstående rett frem ut fra foregående.

$$\boldsymbol{R}_{AC}(t) = e^{-2(\lambda_l + \lambda_{NE})t}(1-(1-e^{-(2\lambda_L + \lambda_{NE})})^2) \qquad (6)$$

d) For det gitte tilfellet når nettelementer og aksesslinker er feilfrie, kan vi sette opp følgende blokkskjema for forbindelsen A, B:



(Merk at uttrykket fra formelsamlingen hvor alle paralellgreiner er like <u>ikke</u> er gyldig.) Dette gir

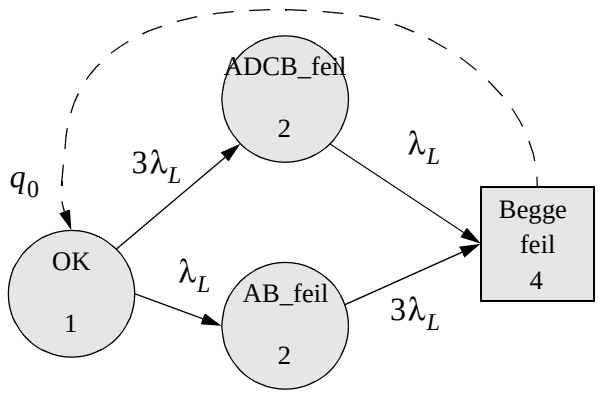
$$\begin{aligned} \boldsymbol{R}_{AB}(t) &= 1-(1-R_L(t))(1-R^3{}_L(t)) \\ &= e^{-\lambda_L t}+e^{-3\lambda_L t}-e^{-4\lambda_L t} \end{aligned} \tag{7}$$

Videre finner vi (fra formelsamling)

$$\mathrm{MTFF}_{AB} = \int_0^\infty \boldsymbol{R}_{AB}(t)dt = \frac{1}{\lambda_L}+\frac{1}{3\lambda_L}-\frac{1}{4\lambda_L} = \frac{13}{12\lambda_L} \tag{8}$$

Alternativ løsning av dette punktet

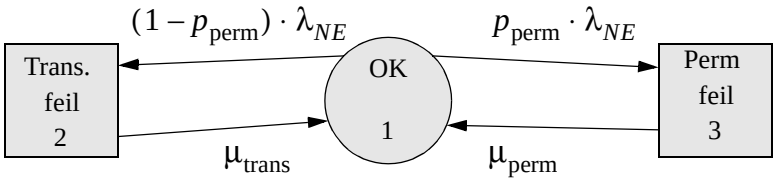
En alternativ fremgangsmåte er å etablere tilstandsdiagram og benytte dette som utgangspunkt for en metodisk løsning som bekrevet i kompendiet



eller ved en middelverdibetraktning

$$\mathrm{MTFF}_{AB} = \frac{1}{4\lambda_L}+\frac{1}{4}\cdot\frac{1}{3\lambda_L}+\frac{3}{4}\cdot\frac{1}{\lambda_L} = \frac{13}{12\lambda_L}$$

e) Tilgjengeligheten til nettelementet, A_{NE}, finnes ved å etablere en tilstandsmodell for elementet og bestemme den stasjonære sannsynligheten for å være i oppetilstand. Merk at antagelsen om kun en feil ad gangen gjør at dette ikke er nødvendig med transisjon mellom 2 og 3.



Setter opp to balanseligninger og normeringsligning for systemet og finner tilstandssansynligheten for OK tilstanden, p_1

$$\begin{aligned} p_1(p_{\text{perm}}\cdot\lambda_{NE}) &= p_3\mu_{\text{perm}} \\ p_1((1-p_{\text{perm}})\cdot\lambda_{NE}) &= p_2\mu_{\text{trans}} \\ p_1+p_2+p_3 &= 1 \end{aligned} \tag{9}$$

$$A_{NE} = p_1 = \frac{\mu_{\text{perm}}\mu_{\text{trans}}}{\mu_{\text{perm}}\mu_{\text{trans}} + (p_{\text{perm}}\cdot\lambda_{NE})\mu_{\text{trans}} + ((1-p_{\text{perm}})\cdot\lambda_{NE})\mu_{\text{perm}}} \tag{10}$$

Alternativ løsning av dette punktet

En alternativ fremgangsmåte som mindre direkte og som krever mindre arbeid er å benytte at

$$A_{NE} = \frac{\text{MUT}_{NE}}{\text{MTBF}_{NE}} = \frac{\text{MUT}_{NE}}{\text{MUT}_{NE}+\text{MDT}_{NE}} \tag{11}$$

og se resultatene fra punkt f) direkte, se likningene (13) og (15) med tilhørende kommentarer. Innsatt i (11) får vi da (10).

(Når $\text{MUT}_{NE} \gg \lambda_{NE}$ så vil $\text{MTBF}_{NE} \approx \text{MUT}_{NE} = 1/\lambda_{NE}$ og vi kan tilnærme

$$A_{NE} \approx 1-\lambda_{NE}\cdot\text{MUT}_{NE} = 1-\lambda_{NE}\cdot\left(\frac{p_{\text{perm}}}{\mu_{\text{perm}}}+\frac{1-p_{\text{perm}}}{\mu_{\text{trans}}}\right) \tag{12}$$

Men det må gis en solid begrunnelse før denne fremgangsmåten kan “aksepteres”.)

f) Siden alle “oppeperioder” er opphold i tilstanden OK og oppholdstidten i dette er negativ eksponensialfordelt (hukommelsesløs) så har vi:

$$\mathrm{MUT}_{NE} = \mathrm{MTFF}_{NE} = \mathrm{MTTF}_{NE} = 1/\lambda_{NE} \tag{13}$$

(For i finne MTFF_{NE} og MTTF_{NE} kan selvsagt den metodiske fremgangsmåten beskrevet i komendiet[1] også benyttes, men den vil kreve unødig arbeid.)

“Standard” fremgansmåte gir:

$$\begin{aligned}\mathrm{MTBF}_{NE} &= \frac{\mathrm{MUT}_{NE}}{A_{NE}} \\ &= \frac{\mu_{\mathrm{perm}}\mu_{\mathrm{trans}} + (p_{\mathrm{perm}} \cdot \lambda_{NE})\mu_{\mathrm{trans}} + ((1-p_{\mathrm{perm}}) \cdot \lambda_{NE})\mu_{\mathrm{perm}}}{\mu_{\mathrm{perm}}\mu_{\mathrm{trans}}\lambda_{NE}} \\ &= \frac{1}{\lambda_{NE}} + \frac{p_{\mathrm{perm}}}{\mu_{\mathrm{perm}}} + \frac{1-p_{\mathrm{perm}}}{\mu_{\mathrm{trans}}}\end{aligned} \tag{14}$$

og følgelig

$$\mathrm{MDT}_{NE} = \mathrm{MTBF}_{NE} - \mathrm{MUT}_{NE} = \frac{p_{\mathrm{perm}}}{\mu_{\mathrm{perm}}} + \frac{1-p_{\mathrm{perm}}}{\mu_{\mathrm{trans}}} \tag{15}$$

Det siste resultatet kan selvsagt også lett utledes ved probabalistiske argumenter. Det tjener som et hint til siste del av oppgaven. Vi ser med at sannsynlighet p_{perm} vil vi ha en negativt eksponensialfordelt nedetid med forventning $\mu^{-1}{}_{\mathrm{perm}}$ og med sannsynlighet $1-p_{\mathrm{perm}}$ vil vi ha en negativt eksponensialfordelt nedetid med forventning $\mu^{-1}{}_{\mathrm{trans}}$. Følgelig får vi en hypereksponensielt fordelt nedetid og:

$$P(T_{D,\,NE} > \tau) = p_{\mathrm{perm}} e^{-\mu_{\mathrm{perm}}\tau} + (1-p_{\mathrm{perm}}) e^{-\mu_{\mathrm{trans}}\tau}$$

1. Gjøre feiltilstander absorberende, slå dem sammen, introdusere hjelpeintensitetenb q_0.

Oppgave 3 Simulering av link brudd og reparasjon

a) *Beskriv i detalj hvordan du kan lage en effektiv trekningsalgoritme ved bruk av invers-transform metoden til å generere T variater som er Weibull-fordelte. Anta at simuleringsspråket/verktøyet kan trekke U variater som er kontinuerlig, uniformfordelte mellom 0 og 1. Hvorfor er invers-transform metoden den mest effektive for Weibull-fordelte variater?*

Inverstransform metoden er den mest effektive trekningsteknikken siden enhver trekning av U gir et T variat. Forutsetningen er at fordelingsfuksjonen kan inverstransformeres. Sjekker nå at dette er mulig for Weibull-fordeling

$$F(x) = 1 - e^{-(\lambda x)^{\gamma}} \tag{16}$$

Sett

$$\begin{aligned} F(x) &= U' \\ 1 - e^{-(\lambda x)^{\gamma}} &= U' \\ e^{-(\lambda x)^{\gamma}} &= 1 - U' = U \\ -(\lambda x)^{\gamma} &= \log U \\ -\lambda x &= (\log U)^{1/\gamma} \\ x &= \frac{-(\log U)^{1/\gamma}}{\lambda} \end{aligned} \tag{17}$$

Hvor $1 - U'$ og U er begge unifromfodelte i [0,1], derfor linje 3 over.

Altså, Weibull fordelingen kan inverstransformeres.

Kommentar: en rekke av besvarelsene løser inverstransformen slik det feilaktig stod i løsningsforslaget til eksamen desember 1999 før dette ble rettet. Dette ble det tatt hensyn til ved sensur av eksamen desember 2001.

b) *Hva er sannsynligheten for at $X \geq 200$? Beregn et 95% konfidensintervall for $\bar{X}$ (kvantiler i normal-fordelingen finner du i tabell 3.1). Hvorfor kan vi anta at $\bar{X}$ er normalfordelt?*

```
                    H I S T O G R A M S
                    *******************

                       S U M M A R Y

TITLE       /   (RE)SET/   OBS/  AVERAGE/EST.ST.DV/ STD.ERR./  MINIMUM/  MAXIMUM
X                 0.000  10000   100.188    50.165     0.502   -81.936   294.938

CELL/LOWER LIM/    N/   FREQ/  CUM :
                                          I-----------------------------
   0 -INFINITY   239    0.02    2.39     I*********
   1     0.000   112    0.01    3.51     I****
   2    10.000   187    0.02    5.38     I*******
   3    20.000   243    0.02    7.81     I*********
   4    30.000   356    0.04   11.37     I*************
   5    40.000   428    0.04   15.65     I****************
   6    50.000   513    0.05   20.78     I*******************
   7    60.000   657    0.07   27.35     I************************
   8    70.000   685    0.07   34.20     I*************************
   9    80.000   792    0.08   42.12     I******************************
  10    90.000   808    0.08   50.20     I*******************************
  11   100.000   814    0.08   58.34     I*******************************
  12   110.000   736    0.07   65.70     I***************************
  13   120.000   678    0.07   72.48     I*************************
  14   130.000   613    0.06   78.61     I************************
  15   140.000   528    0.05   83.89     I******************
  16   150.000   469    0.05   88.58     I*****************
  17   160.000   333    0.03   91.91     I************
  18   170.000   255    0.03   94.46     I*********
  19   180.000   170    0.02   96.16     I******
[190,200) 20   190.000   124    0.01   97.40     I*****
[200,210) 21   200.000   102    0.01   98.42     I****
  22   210.000    74    0.01   99.16     I***
  23   220.000    84    0.01  100.00     I***
                                          I-----------------------------
```

Figur 1 Simuleringsresultat for observatoren X.

Avlest fra figuren: $P(X \geq 200) \;=\; 1 - P(X < 200) \;=\; 1 - 0.974 \;=\; 0.026$.

Ettersom $X \sim$ i.i.d og at vi har et stort antall observasjoner (10000, se rapporten), så kan vi anta at $\bar{X}$ er normalfordelt. Dermed blir 95% konfidenintervallet (dvs. 1-α=0.95, og $\alpha \;=\; 0.05$):

$$[\bar{X} - u_{\alpha/2} S_{\bar{X}}, \bar{X} + u_{\alpha/2} S_{\bar{X}}] \tag{18}$$

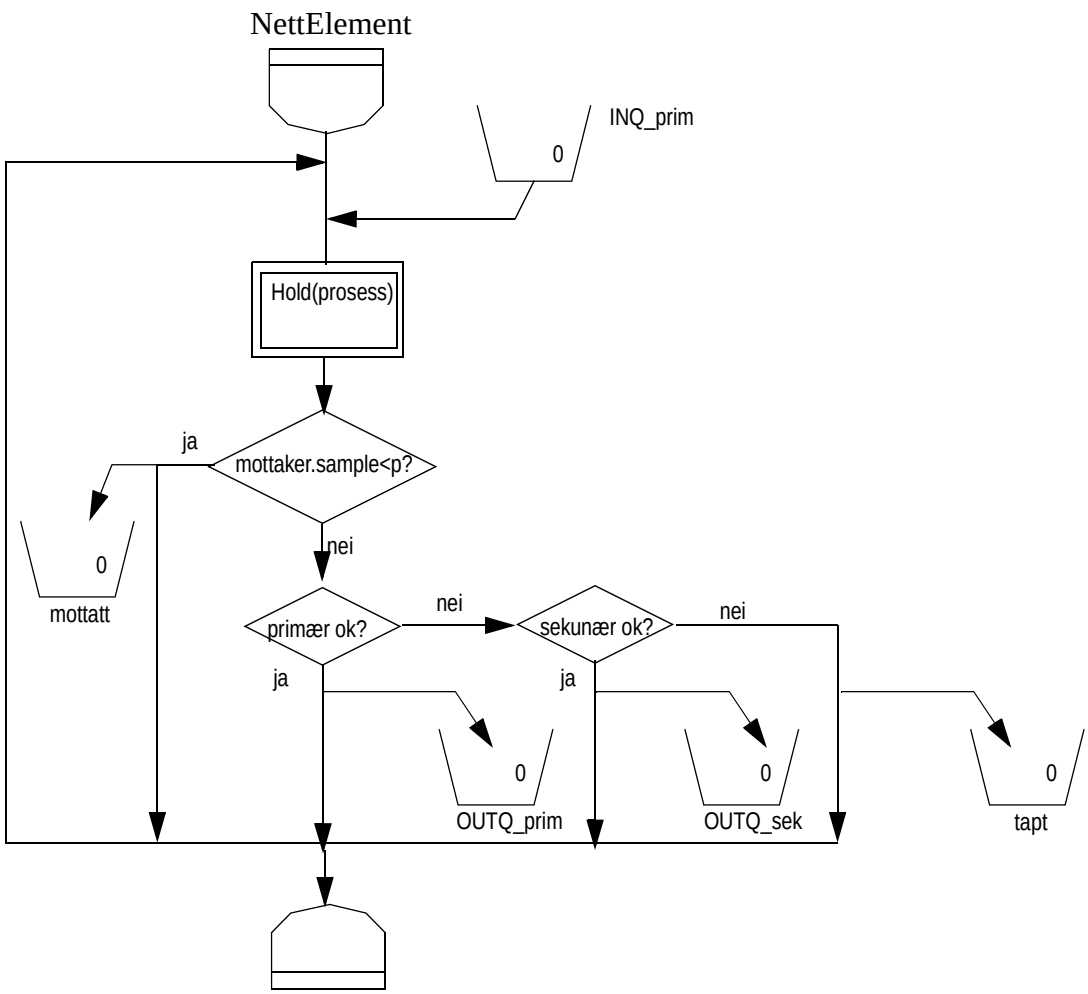
hvor

$$\begin{aligned} \bar{X} \;&=\; 100.188 \\ S_{\bar{X}} \;&=\; 0.502 \\ u_{\alpha/2} \;=\; u_{0.05/2} \;&=\; u_{0.025} \;=\; 1.960 \end{aligned} \tag{19}$$

$$[99.204,\ 101.172] \tag{20}$$

c) Lag en modell av et nett-element som kan lese pakker fra et innkommende buffer (sekundær eller primær link) og sende pakker til utgående buffer (primær- eller sekundær-link), eller til mottaker. Vurder om det er hensiktsmessig med en eller to entitetsklasser for nettelementmodellen.

Aktivitetsdiagram for et NettElement. I denne løsningen vil det instansieres ett NettElement for primær og ett for sekundær ringen



Kommentar: Her ble det lagt vekt på at kandidatene fikk tydelig fram: bruk av inn- og ut-buffer, sjekker status på utgående linker, viser markert forskjell på primær- og sekundær-link, prosessering av pakker, ikke terminerer prosessen etter en pakke (skal ha "evig-løkke"), registrer tapte pakker.

d) Lag en entitetsklasse som modellerer både feiling- og reparasjon av linker med antakelsene som gitt over. Hvordan vil du modellere at et nett-element har feilet?

Kode for link feil og reparasjon (origp er nabonode "oppstrøms", mens origs er "ned-strøms" nabo):

```
    entity CLASS linkfeil(origp,origs);
    REF(NettElement)origp,origs;
    BEGIN
loop:
       Hold(nestefeil.sample);
       origp.prim := FALSE;
       origs.sek  := FALSE;
       Hold(reparer.sample);
       origp.prim := TRUE;
       origs.sek  := TRUE;
       repeat;
    END linkfeil;
```

Nett-element feil modelleres ved å introdusere feil på linkene på begge sider av et nett-elements, f.eks.

```
    entity CLASS nodefeil(origp,origs);
    REF(NettElement)origp,origs;
    BEGIN
loop:
       Hold(nestefeil.sample);
       origp.prim := FALSE;
       origp.sek := FALSE;
       origs.prim := FALSE;
       origs.sek  := FALSE;
       Hold(reparer.sample);
       origp.prim := TRUE;
       origp.sek := TRUE;
       origs.prim := TRUE;
       origs.sek  := TRUE;
       repeat;
    END linkfeil;
```

Kommentar: Besvarelsen krevde ikke kildekode. Denne er tatt med for helhetens skyld.

Komplett kode:

```
BEGIN
   EXTERNAL CLASS demos = "d:\cim\demos\demos";

   demos BEGIN
    REF(histogram)trekk;
    REF(rdist)prosess, mottatt, nestepakke, nestefeil, reparer;
    REF(bin) ARRAY tapt(1:4);
    REF(bin) ARRAY framme(1:4);
    REF(bin) ARRAY primq(1:4);
    REF(bin) ARRAY sekq(1:4);
    REF(NettElement) ARRAY NErefp(1:4);
    REF(NettElement) ARRAY NErefs(1:4);
    INTEGER i;
    REAL prosint, feilint, repint, pakkeint;
```

```
    entity CLASS NettElement(id, inq, outqp, outqs, pi);
    INTEGER id;REF(bin)inq;REF(bin)outqp;REF(bin)outqs;REAL pi;
    BEGIN
       BOOLEAN prim, sek;
       prim:=TRUE;
       sek:=TRUE;
loop:
       ! wait until packet arrived;
       inq.take(1);
       Hold(prosess.sample);
       IF mottatt.sample < pi THEN
          framme(id).give(1)
       ELSE
          BEGIN
         IF prim THEN outqp.give(1)
         ELSE
            BEGIN
          IF sek THEN outqs.give(1)
          ELSE
             tapt(id).give(1);
            END;
          END;
       repeat;
    END NE;

    entity CLASS bruker(outq);REF(bin)outq;
    BEGIN
loop:
       Hold(nestepakke.sample);
       outq.give(1);
       repeat;
    END bruker;

    entity CLASS linkfeil(origp,origs);
    REF(NettElement)origp,origs;
    BEGIN
loop:
       Hold(nestefeil.sample);
       origp.prim := FALSE;
       origs.sek  := FALSE;
       Hold(reparer.sample);
       origp.prim := TRUE;
       origs.sek  := TRUE;
       repeat;
    END linkfeil;

    prosint := 1;
    feilint := 1;
    repint := 0.01;
    prosint := 1;
    feilint := 1;
    repint := 0.01;
    pakkeint := 0.3;
    prosess :- NEW NegExp("Prosess", prosint);
    nestefeil :- NEW NegExp("Linkfeil", feilint);
    reparer :- NEW NegExp("Reparer", repint);
    nestepakke :- NEW NegExp("Mellom pakker", pakkeint);
    mottatt :- NEW Uniform("Mottatt",0,1);

    FOR i:=1 STEP 1 UNTIL 4 DO
```

```
      BEGIN
         framme(i) :- NEW bin(edit("framme",i),0);
         tapt(i) :- NEW bin(edit("tapt",i),0);
         primq(i) :- NEW bin(edit("prim",i),0);
         sekq(i) :- NEW bin(edit("sek",i),0);
      END;

    ! lag topologi: 4 NE i dual-ring;
    ! primaer ring;
    NErefp(1):-NEW NettElement("NE1-p", 1, primq(1), primq(2), sekq(4), 0.30);
    NErefp(1).schedule(0.0);
    NErefp(2):-NEW NettElement("NE2-p", 2, primq(2), primq(3), sekq(1), 0.30);
    NErefp(2).schedule(0.0);
    NErefp(3):-NEW NettElement("NE3-p", 3, primq(3), primq(4), sekq(2), 0.30);
    NErefp(3).schedule(0.0);
    NErefp(4):-NEW NettElement("NE4-p", 4, primq(4), primq(1), sekq(3), 0.30);
    NErefp(4).schedule(0.0);
    ! sekudaer ring;
    NErefs(1):-NEW NettElement("NE1-s", 1, sekq(1), sekq(4), sekq(4), 0.30);
    NErefs(1).schedule(0.0);
    NErefs(2):-NEW NettElement("NE2-s", 2, sekq(2), sekq(1), sekq(1), 0.30);
    NErefs(2).schedule(0.0);
    NErefs(3):-NEW NettElement("NE3-s", 3, sekq(3), sekq(2), sekq(2), 0.30);
    NErefs(3).schedule(0.0);
    NErefs(4):-NEW NettElement("NE4-s", 4, sekq(4), sekq(3), sekq(3), 0.30);
    NErefs(4).schedule(0.0);

    ! start brukere;
    NEW bruker("Bruker 1", primq(1)).schedule(0.0);
    NEW bruker("Bruker 2", primq(2)).schedule(0.0);
    NEW bruker("Bruker 3", primq(3)).schedule(0.0);
    NEW bruker("Bruker 4", primq(4)).schedule(0.0);

    ! lag feil;
    NEW linkfeil("Link 1", NErefp(1), NErefs(2)).schedule(0.0);
    NEW linkfeil("Link 2", NErefp(2), NErefs(3)).schedule(0.0);
    NEW linkfeil("Link 3", NErefp(3), NErefs(4)).schedule(0.0);
    NEW linkfeil("Link 4", NErefp(4), NErefs(1)).schedule(0.0);

    !trace;
    Hold(500);
  END;
END;
```