

# Sammenligning av ulike policier for prioritering ved bruk av Diffservs "forwarding" klasser

**Aso Amedyan**

Master i kommunikasjonsteknologi  
Oppgaven levert: Juli 2009  
Hovedveileder: Norvald Stol, ITEM



# Oppgavetekst

Med utgangspunkt i Diffservs "forwarding" klasser og et antall realistiske trafikk-kilder skal ulike policier/strategier for lokal prioritering sammenlignes. Vi fokuserer på en ingressnode. Trafikk-kildene vil typisk tilhøre de tre hovedklassene telefoni, data og video. Minst en pre-emptiv og en non-pre-emptiv policy vil bli sammenlignet. Hovedfokus i studien vil være å observere effekten av de ulike policier på QoS (tap, forsinkelse og jitter) for de ulike trafikk-kildene/prioritetsklassene. Simulering vil bli brukt for denne studien.

Oppgaven gitt: 25. februar 2009

Hovedveileder: Norvald Stol, ITEM



Norges Teknisk-Naturvitenskapelige Universitet  
Institutt for Teknisk Kybernetikk og telematikk



## **Sammenligning av ulike policyer for prioritering ved bruk av "Diffservs forwarding" klasser**

Sivilingeniørutdanning i  
industrielle datasystemer

av  
Aso Amedyan

## Oppgavetekst

Sammenligning av ulike policyer for prioritering ved bruk av "Diffservs forwarding" klasser

-----

Med utgangspunkt i "Diffservs forwarding" klasser og et antall realistiske trafikkilder skal ulike policyer/strategier for lokal prioritering sammenlignes. Vi fokuserer på en ingressnode. Trafikkildene vil typisk tilhøre de tre hovedklassene telefoni, data og video. Minst en "pre-emptiv" og en "non-pre-emptiv" policy vil bli sammenlignet. Hovedfokus i studien vil være å observere effekten av de ulike policyer på "Quality of Service (QoS)" (tap, forsinkelse og "jitter") for de ulike trafikkildene/prioritetsklassene. Simulering vil bli brukt for denne studien.

## Mål

Utformingen av konkrete mål for oppgaven har vært en dynamisk prosess. Med dette menes at mål og målformuleringer har forandret seg etter hvert som jeg har tilegnet meg ny kunnskap.

Mål for oppgaven er:

1. Studere og få inngående kunnskap om teknologiene. Det vil si at jeg må sette meg inn i hvordan de tre hovedtrafikk kildene (tale, video og data) genereres med tanke på prioritering av pakker som sendes videre til de tre standard klassene " Expedited Forwarding (EF), Assured Forwarding (AF) og Best Effort (BE)" i "Differentated Services (Diffserv)".
2. Skal se på hvilke mekanismer som finnes i "Diffserv packet scheduling".
3. Beskrive konkrete og presise sammenligning av "preemption" og "non preemption" for hvordan disse kan garantere "Quality of Service".
4. Utføre en så realistisk simuleringsstudie som mulig av de policyene som blir brukt i oppgaven

## Forord

Hovedoppgaven som er beskrevet i dette dokumentet inneholder resultater fra diplomarbeidet utført vårsemesteret 2009 av Aso Amedyan. Kandidaten er sivilingeniørstudent ved Institutt for teknisk kybernetikk, industrielle datasystemer, ved Norges teknisk-naturvitenskapelige universitet i Trondheim.

Tekstoppgaven er et samarbeid mellom kandidaten og faglærer ved Institutt for telematikk, som tar sikte på å integrere tjenester som tale, video og data over "Differentiated Services (Diffserv)" i et nett. Det var et forslag fra min veileder, Norvald Stol, om bl.a. å utrede generering av tale, video og data og sette prioritering på disse med hensyn til viktigheten av pakker som sendes videre til brukeren.

Med bakgrunn i min interesse for datakommunikasjon, var dette en oppgave jeg syntes var veldig interessant. Dette området er i sterk utvikling og jeg ville med denne oppgaven gjøre meg mer attraktiv for arbeidsgiveren (NTNU, IT-avdeling).

Jeg vil benytte anledningen til å rette en stor takk til min veileder Norvald Stol (NTNU) for all hjelp, entusiasme og faglig støtte i denne perioden.

Trondheim, 22. Juli 2009

Aso Amedyan



# Innholdsfortegnelse

Oppgavetekst.....	I
Mål .....	II
Forord.....	III
Innholdsfortegnelse .....	IV
Forkortelser.....	VIII
Ordliste .....	IX
Sammendrag .....	1
<b>1 Innledning .....</b>	<b>3</b>
<b>2 Grunnleggende teori.....</b>	<b>4</b>
2.1 Bakgrunn .....	4
2.2 "Differentiated Service (DiffServ)" .....	4
2.3 Per Hop Behavior (PHB) .....	5
2.4 Sammenligning av ulike policyer for prioritering ved bruk av "DiffServ forwarding" klasser .....	6
2.5 Hierarkisk pakkescheduler .....	6
2.6 Fordeler med å gjennomføre "DiffServ" [51] .....	9
2.7 Differensiert Service Code Point (DSCP).....	9
2.7.1 Implementering av kvalitetsservice policyer med DSCP .....	10
2.7.2 DiffServ feltsdiagram.....	11
2.8 Assured Forwarding (AF) PHB .....	12
2.9 Videostreaming.....	13
2.10 "Preemption og non-preemption" .....	14
2.10.1 "Preemptive multitasking" .....	14
2.11 "Token Bucket" modell .....	16
2.11.1 Gjennomsnittlig rate .....	16
2.11.2 "Burst size" .....	17
2.12 "Token Bucket i Differentiated Services" .....	17
2.13 Tjenestekvalitet (QoS) .....	18
2.13.1 Hva er Quality of Service?.....	18
2.14 Ytelse:.....	18
2.15 Synkronisering.....	19
<b>3 Trafikk modell for tale, video og data.....</b>	<b>20</b>
3.1 Simulator modell .....	21
3.2 Trafikkilder: .....	24
<b>4 Generering av tale-, video- og datakilder .....</b>	<b>25</b>
4.1 Matematiske ligninger for tale- og videokilder.....	25
4.2 Metode for generering av kunstige tale [67].....	26
4.3 Beregning av AV/PÅ varighet for e-post og ftp .....	29

4.3.1	<i>E-post</i> .....	29
4.3.2	<i>ftp</i> .....	29
4.4	Beregning av "time spacing" for business telefoni (EF-klasse).....	30
4.4.1	<i>Gjennomsnittlig beregning av overføringstid for tale</i> .....	31
4.6	Beregning av "time spacing" for videokilder (EF- og AF1-klasse) [70]: .....	31
4.6.1	<i>Gjennomsnittlig beregning av overføringstid for video1 og video2 [70]</i> .....	31
4.6.2	<i>Beregning av pakkerate for video1 og video2</i> .....	32
4.7	Beregning av "time spacing" for data1 og data2 (AF2 og BE) henholdsvis e-mail og ftp [71].....	32
4.7.1	<i>Gjennomsnittlig beregning av overføringstid for data1 og data2 [71]</i> .....	33
4.7.2	<i>Beregning av pakkerate for data1 og data 2:</i> .....	33
<b>5</b>	<b>Resultater</b> .....	<b>34</b>
5.1	Beregning av antall pakker for trafikkilder.....	34
5.1.1	<i>Hastighetsberegning for 90 Mbps:</i> .....	34
5.1.2	<i>Hastighetsberegning for 80 Mbps:</i> .....	36
5.1.3	<i>Hastighetsberegning for 95 Mbps:</i> .....	37
5.2	Beregning av middelvei, varians og konfidensintervall for trafikktyper:.....	39
5.3	Registrerte og beregnede resultater i form av figurer og tabeller .....	41
<b>6</b>	<b>Diskusjon og konklusjon</b> .....	<b>49</b>
6.1	Løsningsmetoder for trafikktyper .....	49
6.2	Statistisk registrering .....	51
6.3	Feilkilder.....	51
2.6	Programmeringsspråk .....	51
<b>7</b>	<b>Videre arbeid</b> .....	<b>53</b>
<b>8</b>	<b>Litteraturreferanser</b> .....	<b>54</b>
<b>9</b>	<b>Vedlegg</b> .....	<b>62</b>
9.1	Kode for simuleringen .....	62
9.2	Utskrift av genererte data for kildetyper .....	76
9.3	Vedlagt CD .....	91
9.4	Oppsett og kjøring av programmet .....	91

## OVERSIKT OVER FIGURER

FIGUR 2-1	"ToS" BYTE FØR "DIFFSERV" [47].....	5
FIGUR 2-2	"ToS" BYTE ETTER "DIFFSERV" [47].....	5
FIGUR 2-3	"DIFFSERV" HIERARKISK PAKKESCHEDULER.....	7
FIGUR 2-4	PAKKE BEHANDLING I "DIFFSERV" [16].....	8
FIGUR 2-5	ILLUSTRERER FORSKJELLEN MELLOM "SHAPING" OG "POLICING" [64] .....	9
FIGUR.2-6	ILLUSTRERER "TOKEN BUCKET" MODELL [19].....	16
FIGUR.2-7	PRODUSENT/FORBRUKER AKTIVITETSDIAGRAM [66].....	19

FIGUR 3-8 TILSTANDSDIAGRAM FOR ET SYSTEM MED N TALEKILDER .....	20
FIGUR 3-9 SYSTEM MED TO TILSTANDER.....	20
FIGUR 3-10 SIMULATOR MODELL.....	22
FIGUR 3-11 FORKLARENDE ILLUSTRASJON FOR SIMULATORMODELL.....	23
FIGUR 4-12 TILSTANDSOVERGANGSMODELL FOR SAMTALE [67] .....	28
FIGUR.5-13 DIAGRAMMET VISER SAMMENLIGNING AV ULIKE KILDETYPER FOR "PREEMPTIN" MED HENSYN PÅ FORSINKELSE. BELASTNINGEN ER 90 % MED ULIKE SEED-VERDIER.....	42
FIGUR 5-14 DIAGRAMMET VISER SAMMENLIGNING AV ULIKE KILDETYPER FOR "NON-PREEMPTIN" MED HENSYN PÅ FORSINKELSE. BELASTNINGEN ER 90 % MED ULIKE SEED-VERDIER.....	44
FIGUR 5-15 DIAGRAMMET VISER SAMMENLIGNING AV ULIKE KILDETYPER FOR "PREEMPTIN" MED HENSYN PÅ FORSINKELSE. BELASTNINGEN ER 80 % MED ULIKE SEED-VERDIER.....	46
FIGUR 5-16 DIAGRAMMET VISER SAMMENLIGNING AV ULIKE KILDETYPER FOR "NON-PREEMPTIN" MED HENSYN PÅ FORSINKELSE. BELASTNINGEN ER 80 % MED ULIKE SEED-VERDIER.....	48

## OVERSIKT OVER TABELLER

TABELL 2-1 ILLUSTRERER DSCP OG SPESIFIKASJON AV AF KLASSENE MED SANNSYNLIGHET [33].....	12
TABELL 4-2 GJENNOMSNIITTLIG PERIODER FOR AV/PÅ TALEKILDER [67].....	28
TABELL 4-3 FORKJELLIGE TYPE INFORMASJON FOR DATAGRAM [68] .....	30
TABELL 4-4 TRAFIKKMODELLERS EGENSKAPER [68] .....	30
TABELL4-5 GJENNOMSNIITTLIG PERIODER FOR AV/PÅ (VIDEO1 OG VIDEO2) [70].....	32
TABELL 4-6 GJENNOMSNIITTLIG PERIODER FOR AV/PÅ (FTP OG E-MAIL) [68].....	33
TABELL 5-7 MÅLERESULTATER ETTER 10 GANGER KJØRING AV 10 UAVHENGIG OBSERVASJONER FOR "PREEMPTION" MED 90 % BELASTNING. ....	41
TABELL 5-8 VISER KVADRERTE VERDIER AV MÅLERESULTATER FOR "PREEMPTION" MED 90 % BELESTNING. ....	41
TABELL 5 9 VISER BEREKNING FOR KONFIDENSGRENSER BASERT PÅ 10 UAVHENGIG OBSERVASJONER FOR "PREEMPTION" MED 90 % BELASTNING. ....	42
TABELL 5-10 MÅLERESULTATER ETTER 10 GANGER KJØRING AV 10 UAVHENGIG OBSERVASJONER "FOR NON- PREEMPTION" MED 90 % BELASTNING. ....	43
TABELL 5-11 VISER KVADRERTE VERDIER AV MÅLERESULTATER FOR "NON-PREEMPTION" MED 90 % BELESTNING. ....	43
TABELL 5-12 VISER BEREKNING FOR KONFIDENSGRENSER BASERT PÅ 10 UAVHENGIG OBSERVASJONER FOR "NON- PREEMPTION" MED 90 % BELASTNING. ....	44
TABELL 5- 13 MÅLERESULTATER ETTER 10 GANGER KJØRING AV 10 UAVHENGIG OBSERVASJONER FOR "PREEMPTION" MED 80 % BELASTNING. ....	45
TABELL 5 14 VISER KVADRERTE VERDIER AV MÅLERESULTATER FOR "PREEMPTION" MED 80 % BELESTNING. ....	45
TABELL 5-15 VISER BEREKNING FOR KONFIDENSGRENSER BASERT PÅ 10 UAVHENGIG OBSERVASJONER FOR "PREEMPTION" MED 80 % BELASTNING. ....	46
TABELL 5-16 MÅLERESULTATER ETTER 10 GANGER KJØRING AV 10 UAVHENGIG OBSERVASJONER FOR "NON- PREEMPTION" MED 80 % BELASTNING. ....	47
TABELL 5-17 VISER KVADRERTE VERDIER AV MÅLERESULTATER FOR "NON-PREEMPTION" MED 80 % BELESTNING. ....	47

---

TABELL 5-18 VISER BEREGNING FOR KONFIDENSGRENSER BASERT PÅ 10 UAVHENGIG OBSERVASJONER FOR "NON-  
PREEMPTION" MED 80 % BELASTNING. ....48

---

## Forkortelser

AF	Assured Forwarding
BE	Best Effort
DEMOS	Discrete Event Modelling On Simula
dp	drop precedence
DS	Differentiated Services
DSCP	Differensiert Service Code Point
EF	Expedited Forwarding
GB RAM	Gigabyte Random Access Memory
GMPLS	Generalized Multi-Protocol Label Switching
IP	Internet Protocol
IT	Information Technology
Kbps	Kilobit per sekund
Mbps	Megabit per sekund
MPLS	Multi-Protocol Label Switching
PHB	Per Hop Behavior
QoS	Quality of Service
RFC	Request for Comments
SIP	Session Initiation Protocol
SIMULA	simulation language
SLA	Service Level Agreement
TB	Token Bucket
Tdt	double talk duration
Tms	mutual silence duration
Tst	single talk duration
TV	Television
simtid	simuleringstid
UDD	unconstrained delay data

---

## Ordliste

- FTP** *File Transfer Protocol*. En TCP basert, transaksjonsorientert fil overføringsprotokoll brukt i TCP/IP nettverk, spesielt Internet.
- GB** Gigabyte (GB) er måleenhet for datalagringskapasitet. En gigabyte er en enhet for informasjon eller datalagringskapasitet, og betyr enten nøyaktig én milliard bytes ( $1000^3$  eller  $10^9$ ) eller omtrent 1,07 milliarder bytes ( $1024^3$ ). For å minske forvirringen rundt dette er det innført en enhet gigabyte som alltid betyr 1 073 741 824 ( $1024^3$  eller  $2^{30}$ ) bytes [75].
- IP** *Internet Protocol*. Nettverklagsprotokollen i TCP/IP stakken som blir brukt i Internet.
- ITU-T** *International Telecommunication Union Telecommunications*. ITU-T er et internasjonalt organ med medlemsland, der oppgaven er å definere anbefalinger og standarder relatert til den internasjonale telekommunikasjonsindustrien.
- Jitter** En forvrengning av et signal som forplanter seg gjennom nettverket, hvor signalet varierer fra dens originale referansetid. I pakkesvitsjede nettverk, er jitter forvrengningen av tiden mellom pakkene, i forhold til det den ble sendt med.
- NS2** NS eller nettverket simulator (også kalt ns-2, i referanse til nåværende generasjon) er en diskret hendelse nettverk simulator[57].
- QoS** *Quality of Service*. QoS er definert for hver forbindelse på en ende-til-ende basis, med hensyn på de følgende attributtene på en ende-til-ende ATM forbindelse: CLR, Max-CTD og PTP-CDV.
- RAM** Random Access Memory. (RAM) er en type lagringsplass brukt i datamaskiner, som gir tilgang til alle de lagrede dataene i vilkårlig (en. random) rekkefølge.

- TCP      *Transmission Control Protocol*. TCP er en pålitelig, forbindelse- og byteorientert transportlags protokoll innen TCP/IP protokoll rekken.
- TB        "Token Bucket" er en trafikk formingsmekanisme, der muligheten til å sende pakker fra en gitt flyt er kontrollert av tokenener. For pakker som hører til en bestemt flyt skal transmitteres, må tokenet være til stede i bøtten.
- ToS      "Type of Service byte" i IP-hodet er delt inn i tre deler: Precedence feltet (high-order 3 biter), et felt som vanligvis kalles " Type of Service eller TOS" (neste 4 biter), og en reservert bit (laveste bit) [59].

## Sammendrag

Hovedoppgaven som er beskrevet i dette dokumentet er i samarbeid mellom kandidaten og faglærer våren 2009. Forfatterens utgangspunkt var å bruke to protokoller " Multi-Protocol Label Switching (MPLS) og Generalized Multi-Protocol Label Switching (GMPLS)" over "Differentiated Services (diffserv)" i ett nett, men etter videre arbeid med litteraturstudie ble planen forandret. Oppgaven går istedenfor på generering av tale, video og data og sette prioritering på disse med hensyn til viktigheten av pakkene over "DiffServ".

"Diffserv" er en teknikk som brukes for å autorisere og merke pakker med avtalt krav til kvalitet på kanten av nettet og så skal de sentrale nodene bare gjøre ruting og køhåndtering for de forskjellige klassene. Merking gjøres i det eksisterende "Type of Service-feltet" i IP-pakken som nå blir redefinert. Hvilke tjenester og kvaliteter som tilbys, blir nettleverandøravhengig, men arkitekturen legger opp til noen grunnmekanismer [56].

Video konferanser, fjernundervisning, fjernstyringsoperasjon for pasienter med tale og video og mange andre tjenester har blitt mer og mer attraktiv via internett. Prioritering mellom datapakker har blitt sentralt tema inn i teknologien. Gode og effektive video og telekommunikasjoner gir et helt avgjørende bidrag til å kunne realisere bruken av IT på tvers av nasjonalt og internasjonal grenser av ulike typer.

"Backbone" leverandører har i løpet av de siste årene bygget opp forskjellige typer nett for overføring av tale, video og data. Det virker nå som om den teknologiske utviklingen innen internett-teknologier kan muliggjøre en integrasjon av ovennevnte tjenester ved å benytte pakke sortering og prioritering.

Med utgangspunkt i generering av trafikkilder, er det viktig at dette kravet til en hver tid gjenspeiler de forventninger brukere har. Det er derfor viktig å få svar på hvordan "Quality of Service (QoS)" kan garanteres og hvordan tjenestekvalitet kan ivaretas med hensyn på tap, forsinkelse og "jitter".

En enkel definisjon på QoS, er at QoS er et mål på hvor godt nettet oppfører seg og et forsøk på å definere karakteristikk og egenskaper til spesielle tjenester [1] [6]. I dette inngår også muligheten for å differensiere mellom trafikk- eller tjenestetyper, slik at pakkene kan behandles med hensyn på prioritering.

Oppgaven ble følgelig ganske omfattende og jeg har underveis vært nødt til å begrense oppgaven. Jeg har i samarbeid med veileder, kommet frem til følgende punkter.



- Med utgangspunkt i "Diffservs forwarding" klasser og et antall realistiske trafikkilder ble ulike policyer/strategier for lokal prioritering sammenlignet.
- Det er blitt fokusert på en ingressnode. Trafikk-kildene vil typisk tilhøre de tre hovedklassene telefoni, data og video. Minst en "preemptiv" og en "non-preemptiv" policy har blitt sammenlignet.
- Hovedfokuset i studien har vært å observere effekten av de ulike policyer på QoS (tap, forsinkelse og "jitter") for de ulike trafikkildene/prioritetsklassene.
- Dokumentet beskriver i tillegg en simulering som er benyttet for denne studien.

## 1 Innledning

I de siste årene har vi fått raskere og raskere internett tilkoblinger og "backbone" nettet er blitt godt utstyrt med fiber. Man vil nå kunne bruke nettet til å komme med bedre sikkerhet med hensyn på pakketap, forsinkelse og "jitter" for å sende brev, dokumenter og video over internett og mobilnett. Man kan da sette ulike prioriteringer for ulike behov; et eksempel er business-telefoni (typisk telefoni for et aksjemeglerhus). Et annet eksempel på at prioritering av pakker kan lønne seg, er i videokonferanser[9] hvor det stilles høye krav til forsinkelser og synkronisering. Det er da veldig viktig å få de riktige pakkene igjennom slik at deltagerne vil kunne kommunisere på en effektiv måte.

Etter hvert vil kanskje selskaper kunne konkurrere med kabel-TV og vanlig TV. De vil da kunne tilby fullstendig brukerinteraktiv slik at forbrukeren kan velge både program og tidspunkt for underholdningen. Man kan også sende vanlige TV-programmer [8] og serier og gi et komplett TV-tilbud. Siden vi per dags dato har veldig varierende båndbredde til sluttbrukeren, er vi nødt til å finne metoder som kan kompensere for dette slik at flest mulig vil kunne benytte seg av tilbudet med multimedia over nettet.

Når det gjelder data, kan man skille mellom for eksempel en database som distribuerer data og en database som kopierer data.

Men for lyd og video trenger man andre mekanismer for å skille mellom sanntids og ikke sanntids, kommunikasjon, høy prioritet og ikke prioritet som for eksempel (video (via satellitt) og lyd (via telefon) eller bruk av video fjernstyring av roboter i forbindelse med kikkhullskirurgi.

I denne oppgaven skal trafikker (tale, video og data) genereres over en "Differentiated Services (DiffServ)".

---

## 2 Grunnleggende teori

### 2.1 Bakgrunn

Denne hovedoppgaven har i første rekke vært en litteraturstudie. I perioden 26. februar til 24. mars, har jeg holdt på å sette meg inn i protokollene "Multi-Protocol Label Switching (MPLS)" og "Generalized Multi-Protocol Label Switching (GMPLS)" over "Differentiated Services (DiffServ)" samt å undersøke av "Network Simulator 2 (NS2)" som et alternativ for simulering av oppgaven. På forhånd hadde jeg noe kunnskap om MPLS og GMPLS fra høstprosjektet (Aso Amedyan høstprosjekt 2008, NTNU) [10]. Jeg hadde også noe kunnskap om hva "QoS" innebar fra tidligere studie.

"Diffserv" over "MPLS/GMPLS" som skulle være fortsettelse av høstens prosjektoppgave, ble ikke valgt. Men isteden valgte vi (jeg og veileder Norvald Stol) med utgangspunkt i "Diffserv forwarding" klasser å sammenligne tre hovedtrafikkilder (tale, video og data) med ulike policyer/strategier for lokal prioritering.

Jeg begynte med å undersøke "QoS" muligheter ved å bruke "DiffServ" mekanisme. Deretter begynte jeg med å se på forskjellige prioriteringer av trafikkilder.

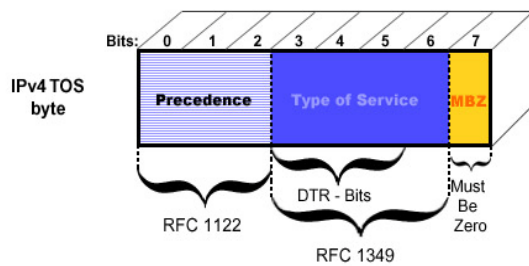
Før jeg begynte med analyse av hvordan teknologiene kan garantere "QoS", satte jeg meg inn i publiserte definisjoner av "QoS", for å danne meg en oppfatning av hva begrepet omfatter [2].

Etter analyse av "QoS", begynte jeg forberedelsene til testdelen. Her satte jeg meg inn i hvordan man legger prioritering til forskjellige klasser med hensyn på viktigheten av pakker som skal videresendes, samt forskjellige policyer "preemption" og "non preemption" [11].

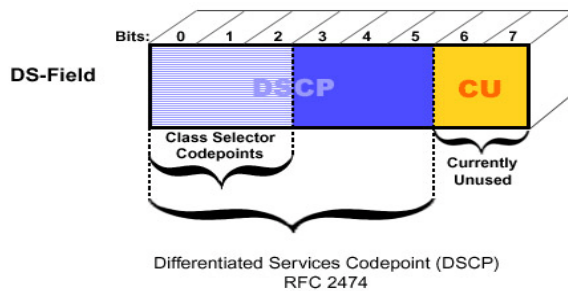
### 2.2 "Differentiated Service (DiffServ)"

"DiffServ" er et rammeverk for å levere QoS gjennom to hovedkomponenter: pakke klassifisering og "Per Hop Behavior (PHB)". Disse komponentene tilrettelegger partisjoneringsnettrafikk i flere prioritetsnivåer eller klasser, slik at hver tjeneste som hører til en bestemt tjenesteklasse får samme videresendingsbehandling.

Pakke klassifisering: Pakke klassifisering og merking skjer via "Differensiert Service Code Point (DSCP)" verdier. DSCP består av seks biter av differensiert service-feltet, som erstatter "Type of Services (ToS) byte" [45] i pakken (se figur 2-1 og 2-2) [47]. Denne seks-biter DSCP støtter opptil 64 (2<sup>6</sup>) distinkte service klasser [48] (RFC 2474).



Figur 2-1 "ToS" byte før "DiffServ" [47]



Figur 2-2 "ToS" byte etter "DiffServ" [47]

### 2.3 Per Hop Behavior (PHB)

I "DiffServ", PHB samler IP "framer" med tilsvarende anvendelseskrav ved å bruke samme "Differentiated Services Code Point (DSCP)" innenfor IP-ramme. De samlede rammer går i samme retning over en gitt link. Deretter prioriteres disse rammene mot hverandre basert på tilgjengelige nettressurser, for eksempel buffer plass, båndbredde tilgjengelighet eller mot trafikk egenskaper som forsinkelse og tap [46].

PHB definerer videresending av pakker tilknyttet en bestemt klasse for trafikk. Dette gjør at nettverksenheten prioriterer trafikk tilsvarende. Det er fire PHB standardgrupper som brukes i "DiffServ" [49], men i denne oppgaven brukes følgende:

1. "Expedited Forwarding (EF) – Premium", høyeste prioritet trafikk.
2. "Assured Forwarding (AF)" – Flere AF klasser med "precedence".
3. "Best effort (BE)" som standard.

---

## **2.4 Sammenligning av ulike policyer for prioritering ved bruk av "DiffServ forwarding" klasser**

I løpet av de siste årene har mange prøvd å utvikle mekanismer for internett for å støtte forskjellige krav av forskjellige type trafikk. "DiffServ" er en av de viktigste mekanismene som er i stand til å oppnå denne tjenesten.

Et eksempel på bruk av "DiffServ" er å dele trafikken inn i ulike klasser og gir ulike prioritering til de ulike klassene, på den måten er "DiffServ" velegnet for bruk i IP-telefoni [13][14]. For å realisere denne teknologien i forbindelse med "Session Initiation Protocol (SIP)" trenger proxyserveren<sup>1</sup> å samarbeide med en enhet for båndbreddekontroll samt ruterne i nettverket [15]. Løsningen er derfor mest nyttig når en tilbyder ønsker å tilby taletjenestene en bedre kvalitet enn for datatjenestene. Dette skyldes at rene datatjenester som regel ikke er så sensitive for forsinkelser som telefonitjenester.

## **2.5 Hierarkisk pakkescheduler**

"Differentiated Services (DiffServ)" gir mulighet til å imøtekomme forskjellige krav til tjenestekvalitet knyttet til ulike kliner og nettverksbrukere. [39]

Det brukes også for å autorisere og merke pakker med bestemt krav til kvalitet på kanten av nettet, og så skal de sentrale nodene bare gjøre ruting og køhåndtering for de forskjellige klassene. Merking gjøres i det eksisterende "Type of Service"-feltet i IP-pakken som nå blir redefinert.

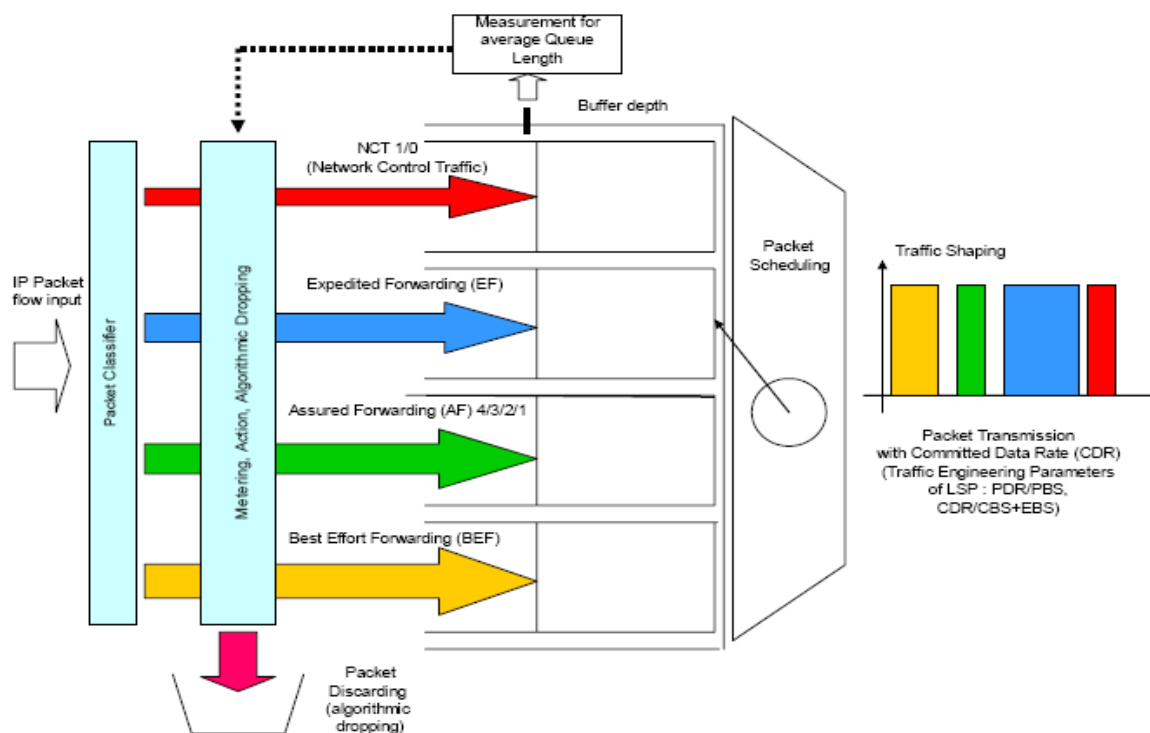
Hvilke tjenester og kvaliteter som tilbys blir avhengig av nettleverandører, men arkitekturen legger opp til noen grunnmekanismer som EF (expedited forwarding) og AF (assured forwarding). EF (expedited forwarding) tar hensyn til forsinkelseskritiske applikasjoner som lyd mens AF (assured forwarding) er mer rettet mot garantert båndbredde [40] [41].

---

<sup>1</sup> En "proxy-server" er en datamaskin som fungerer som et mellomledd mellom en webleser (for eksempel Internet Explorer) og Internett. Proxy-servere bidrar til å forbedre webytelsen ved å lagre en kopi av websider som ofte blir brukt. Når en webleser sender en forespørsel etter en webside som er lagret i "proxy-serverens" samling (serverens hurtigbuffer), sendes den direkte fra "proxy-serveren", noe som går raskere enn å hente den ned fra Internett. Proxy-servere bidrar også til å bedre sikkerheten ved å filtrere ut bestemt webinnhold og skadelig programvare. Proxy-servere brukes mest i nettverk i organisasjoner og firmaer. Personer som kobler til Internett hjemmefra, bruker vanligvis ikke en "proxy-server" [60][61].

Med differensiert Services, forsøker nettet å levere en bestemt type tjeneste basert på QoS spesifisert i hver pakke. Denne spesifikasjonen kan skje på ulike måter, for eksempel ved hjelp av 6-bits "Differentiated Services Code Point (DSCP)" (for nærmere beskrivelse 2.7) i IP-pakker eller kilde- og måladresse. Differensierte tjenester er vanligvis nødvendige for å samle datastrømninger slik at den gir et relativt grovt nivå av trafikk klassifisering [42].

Pakkescheduler bestemmer hvilken pakke som skal sendes videre. Pakken velges på grunnlag av relativ tildeling av tilgjengelig båndbredde iblant "DiffServ" klasser. Dette kan konfigureres ved hjelp av pakkescheduler som kombinerer prioritert-baserte scheduler og weight-baserte scheduler. I prioritert-baserte pakkescheduler, er pakkestrømninger med høyere prioritering privilegert til å bli betjent til tross for trafikk med lavere prioritering. Det er derfor viktig å kontrollere strømmingen av høyere prioritert trafikk, slik at den totale båndbredden ikke blir overbelastet. I weight-baserte pakkescheduler vil hver pakke få tildelt en båndbredde som er proporsjonal med pakkens størrelse. Figur 2-3 illustrerer en hierarkisk fremstilling av pakkescheduler. Figuren viser at strømming av pakker er klassifisert slik at hver klasstypen får tildelt en prioritert-basert pakkescheduler, mens AF-klasse får en rate-basert scheduler [29].

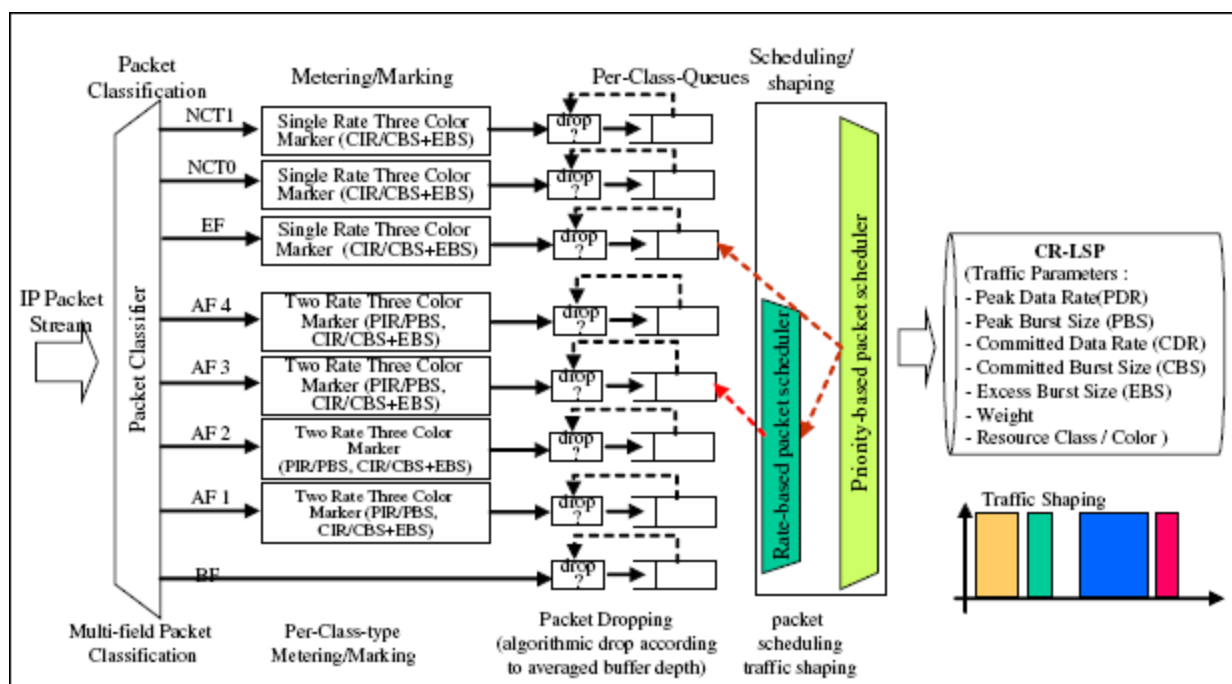


Figur 2-3 "DiffServ" hierarkisk pakkescheduler

- "Assured Forwarding" (AF): AF er en del av "Differentiated Services (DiffServ)" som er en arkitektur som skal gi støtte for ulike tilbud til nettverks-strømmer [18]. En av karakteristikene til "DiffServ" er at den samler strømmer i grupper og behandler alle strømmene i gruppen som

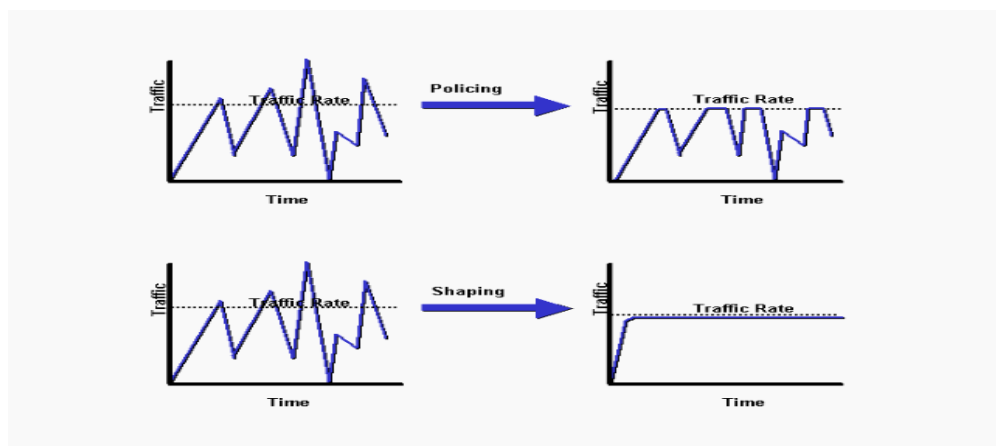
en strøm. AF støtter leveranse av "Internet Protocol (IP)" pakker i fire forskjellige uavhengige AF klasser for videresending (se figur 2-4). Innenfor hver AF klasse kan en IP pakke bli tilordnet en av tre forskjellige grader av "Drop Precedence". AF har opptil 4 forskjellige klasser vi kan klassifisere pakkene i, hvor hver klasse har forskjellige ressurser som båndbredde og bufferplass. Det AF skal gjøre er å sikre forskjellige grader av sikkerhet for at vi får videresendt en pakke. Dette vil gjøre at vi har større sjanse for at vi får igjennom en pakke hvis vi setter pakken i en klasse som har mer båndbredde eller bufferplass, enn i en som har mindre. Det er også forskjellige regler for når vi skal kaste en pakke, hvis vi har overbelastning så vil pakkens AF klasse som pakke være avgjørende for om vi skal kaste pakken eller ikke [27] [16].

- "Expedited Forwarding (EF)": EF er også en del av "DiffServ" og kan bli brukt for å bygge en ende til ende system som har lite tap, lite forsinkelse og lite "jitter". Pakketap, forsinkelse og "jitter" henger alle sammen med køer som kommer som følge av sending av pakker i et nett. Køer kommer som regel fra at innraten til en node er større enn utraten. Det vi prøver med EF, er å minimere køene. Dette skjer ved at vi prøver å finne et veldefinert minimum utsendingsrate, hvor vi med veldefinert mener at den er uavhengig av den dynamiske tilstanden til noden. EF krever at senderaten skal være mindre enn minimum utraten. Dette skjer ved hjelp av forskjellige metoder som for eksempel "shaping" [63] eller "policing" [62] (se figur 2-5). Når vi minimerer køene, kan vi lett si hvor lang tid det vil ta og sende pakker igjennom nettverket. Ulempen er at det ikke er så lett å skalere med EF [21].



Figur 2-4 pakke behandling i "DiffServ" [16]

Den differensierte tjenestearkitekturen er basert på en enkel modell der trafikken inn på et nett er klassifisert og betinget ved grensene til nettet [12].



Figur 2-5 ILLUSTRERER FORSKJELLEN MELLOM "SHAPING" OG "POLICING" [64]

## 2.6 Fordeler med å gjennomføre "DiffServ" [51]

"DiffServ" brukes for ende-til-ende "Quality of Service" funksjoner til å gjennomføre differensierte tjenestearkitekturer. Fordelene med å implementere differensierte tjenester omfatter følgende [51]:

1. Reduserer belastning på nettverksenheter og skaleres lett etter hvert som nettet vokser.
2. Tillate kunder å beholde eksisterende "Layer 3 Type of Service (TOS)"<sup>2</sup> prioriteringsordning som kan være i bruk.
3. Tillate kunder å mikse "DiffServ-compliant" enheter med eventuelle eksisterende ToS-aktivert utstyr i bruk.
4. Reduserer flaskehalsen gjennom effektiv forvaltning av eksisterende bedriftsnettverksressurser.

## 2.7 Differensiert Service Code Point (DSCP)

De seks viktigste bitene av "DiffServ" feltet kalles DSCP. De to siste "Currently Unused" (Cu) biter i "DiffServ" feltet ikke var definert innenfor "DiffServ" arkitekturen. Disse er nå brukt som

<sup>2</sup> Refereres til lag 3 i "Open Systems Interconnection (OSI)" – modellen [52][53].



"Explicit Congestion Notification" (ECN) biter. Rutere på kanten av nettet klassifiserer pakker og merker dem enten med IP "precedence" eller DSCP verdi i et "Diffserv" nett.

DS5	DS4	DS3	DS2	DS1	DS0	ECN	ECN
-----	-----	-----	-----	-----	-----	-----	-----

DSCP er seks biter (DS5-DS0).

ECN er to biter.

Den standardiserte "DiffServ" pakkefeltet er merket med en verdi slik at pakken får en spesiell "forwarding" eller PHB i hver nettnode. Standarden for DSCP er 000 000.

### 2.7.1 Implementering av kvalitetsservice policyer med DSCP

Det store valget er en avveining av tre muligheter: lav forsinkelse, høy pålitelighet og høy gjennomstrømming. Følgende viser spesifikasjon av 8 biter servicetype som er definert ved RFC 791 [54]:

0	1	2	3	4	5	6	7
Precedence			D	T	R	0	0

Bits 0-2: "Precedence".

Bit 3: 0 = Normal forsinkelse, 1 = Lav forsinkelse.

Bit 4: 0 = Normal gjennomstrømming, 1 = høy gjennomstrømming.

Bit 5: 0 = Normal pålitelighet, 1 = høy pålitelighet.

Bit 6-7: Reservert for fremtidig bruk

I "Differentiated Service (DiffServ)" behandles trafikken av mellomliggende systemer med relative prioriteringer basert på "Type of Services (ToS)" feltet. Dette er definert i RFC 2474 og RFC 2475. "DiffServ" erstatter originale spesifikasjonen for pakkeprioritering som er beskrevet i RFC 791.

"Type of Service" gir en indikasjon på abstrakte parametere av kvaliteten på den tjenesten som ønskes. Disse parametrene blir brukt til å lede et valg av aktuelle tjenester når et "datagram" overføres gjennom et spesifikt nett. Flere nett tilbyr "service precedence" som på en eller annen måte behandler høy prioriteringstrafikk som er viktigere enn annen trafikk (generelt ved å godta bare trafikk over en viss prioritet på et tidspunkt ved høy belastning) [54].

## 2.7.2 DiffServ feltsdiagram

1	0	1	0	0	0	ECN	ECN
---	---	---	---	---	---	-----	-----

"DiffServ" standarden benytter samme "precedence" biter (de viktigste bitene er DS5, DS4 og DS3) for prioritetsinnstillinger, men den illustrerer ytterligere definisjoner som tilbyr bedre detaljenivå gjennom bruk av de neste tre bitene i DSCP[33]. "DiffServ" omorganiserer og gir et annet prioriteringsnivå (fortsatt definert ved de tre mest betydelige biter av DSCP) inn i disse kategoriene [33].

Dette systemet prioriterer trafikken først etter klassen, og deretter differensierer og prioriterer samme klassen etter "drop" sannsynlighet. "DiffServ" standarden angir ikke en presis definisjon av lav-, medium- og høydrops sannsynlighet. Det er ikke alle enheter som gjenkjenner "DiffServ" (DS2 og DS1) innstillinger, selv når disse er anerkjent. De vil ikke nødvendigvis utløse samme PHB videresending på hver nettverksnode. Hver node implementerer egne svar basert på hvordan den er konfigurert [33].

"Assured Forwarding" PHB garanterer en viss båndbredde til hver AF klasse og gir tilgang til ekstra båndbredde, hvis det er tilgjengelig. Det er fire AF klasser (AF1x gjennom AF4x). Innenfor hver klasse er det tre "drop" sannsynligheter som er avhengig av nett-policyer. Pakker kan velges for en PHB basert på nødvendig gjennomstrømming, forsinkelse, "jitter", tap eller i henhold til prioritering av tilgang til nettverkstjenester (RFC 2597).

Tabell 2-1 illustrerer DSCP koding og spesifisering av AF klassene med sannsynlighet. Biter DS5, DS4 og DS3 definerer klassen og biter DS2 og DS1 spesifiserer "drop" sannsynligheten, mens bit DS0 alltid er null [33].

**Tabell 2-1 illustrerer DSCP og spesifikasjon av AF klassene med sannsynlighet [33].**

Dro	Klasse 1	Klasse 2	Klasse 3	Klasse 4
Lav Drop Precedence	001010	010010	011010	100010
	AF11	AF21	AF31	AF41
	DSCP 10	DSCP 18	DSCP 26	DSCP 34
Medium Drop Precedence	001100	010100	011100	100100
	AF12	AF 22	AF32	AF42
	DSCP 12	DSCP 20	DSCP 28	DSCP 36
Høy Drop Precedence	001110	010110	011110	100110
	AF13	AF23	AF33	AF43
	DSCP 14	DSCP 22	DSCP 30	DSCP 38

"Expedited Forwarding" (EF) PHB kan brukes til å bygge et lite tap, lav forsinkelse, lav jitter, forsikret båndbredde og ende-til-ende tjenester gjennom "Diffserv" domene. Slike tjenester vises til endepunkter som punkt-til-punkt-forbindelse eller et "virtual Leased Line"<sup>3</sup>. Denne tjenesten har også blitt beskrevet som "premium service". "Codepoint" 101110 er anbefalt for EF PHB som tilsvarer en DSCP verdi av 46 (RFC 2598).

## 2.8 Assured Forwarding (AF) PHB

"Differentiated Service (DiffServ)" er en av flere tjenester som kan tilfredsstillte ulike QoS krav. "DiffServ" leverer en bestemt type tjeneste basert på QoS som er spesifisert i hver pakke. Denne spesifikasjonen kan skje på ulike måter, for eksempel ved hjelp av 6-bits "Differentiated Service Code Point (DSCP)" i IP-pakker eller kilde og destinasjonsadresse [34].

AF PHB definerer fire AF klasser: AF1, AF2, AF3 og AF4. Hver klasse er tildelt en bestemt mengde bufferplass og grensesnitt båndbredde, ifølge "Service Level Agreement (SLA)" med serviceleverandøren eller "policy map".

Det er tre "drop precedence (dp)" verdier: 1, 2 og 3 innenfor hver klasse.

"Assured Forwarding PHB" kan uttrykkes som AF<sub>n</sub>y (RFC 2597) hvor n representerer AF klasses tall (1, 2, 3, 4) og y representerer "drop precedence (dp)" verdi (1, 2 og 3) innenfor AF klasser.

<sup>3</sup> Virtual Leased Line (VLL) er en måte å gi Ethernet basert punkt til punkt kommunikasjon over IP nett.

---

Hvis det er pakkeoverbelastning i en "DiffServ" klasse, for eksempel AF1, kan pakkene i AF1 klassen droppes i henhold til følgende retningslinjen [34]:

$$dp(AFny) \geq dp(AFnz) \geq dp(AFnx)$$

Der "dp" (AFny) er sannsynligheten for at pakker av AFny klasse vil bli slettet. Med andre ord, y betegner "dp" innenfor en AFn klasse. Tabell 2-1 viser DSCP verdier med "dp" for hver klasse.

I følgende eksempel droppes pakker i AF13 klasse før pakker i AF12 klasse, noe som igjen vil bli droppet før pakker i AF11 klasse [34]:

$$dp(AF13) \geq dp(AF12) \geq dp(AF11)$$

Ifølge RFC 2597 kan en AFx klasse bli betegnet ved DSCP xyzab0, der xyz kan være 001, 010, 011 eller 100, og ab representerer "dp" biter [34].

Med kringkasting eller "Live webcasting" kan mottaker bare se og lytte, eksampler er å se på TV eller høre på radio. Det er enveis kommunikasjon fra datasenderen til mottakeren. Mottakeren har ingen adgang til de dataene eller medier som strømmer inn [35].

For "Interaktiv Streaming" kan begge parter sende og motta datastrømmer samtidig. Man kan kommunisere med direkte avspilte data via "streaming" av en video, lyd eller tekst. Alt dette skjer i sanntid og live [35].

## 2.9 Videostreaming

"Videostreaming" kan benyttes til mange formål. Videokonferanse innen kirurgi kan for eksempel benyttes til undervisning, møter, faglig veiledning og i framtiden kanskje også innen behandling (fjernstyring av roboter i forbindelse med kikkhullskirurgi<sup>4</sup>) [55]. I forbindelse med dialyse er videokonferanse benyttet til fjernvisitt for bedre pasientoppfølging og til bedre integrasjon av ulike organisasjonsenheter ved å skape en følelse av en felles arbeidsplass gjennom fjernvisitt, møter og undervisning. Vi har fokus på to type video i denne oppgaven. Den ene er kringkasting eller "Live webcasting" som for eksempel "Live TV" eller avspilling av lagret video, og den andre er "interaktiv streaming" hvor begge parter kan motta og sende datastrømmer samtidig. "Interaktiv streaming" er mer sanntidsvideo. Sanntidsvideo stiller generelt de strengeste kravene til tjenestekvalitet, mens avspilling av lagret video stiller andre og lavere krav. Lagret video kan filoverføres eller "streames", og dette stiller igjen ulike krav til tjenestekvalitet.

---

<sup>4</sup> Det har allerede vært gjort forsøk med dette [55].

---

## 2.10 "Preemption og non-preemption"

I prioritet basert scheduling, kan en høyere prioritet prosess avbryte en lavere prioritet prosess. I et "preemptive" system blir det umiddelbart "switching" over til en høyere prioritert prosess. Alternativt vil en lavere prioritert prosess få lov til å fullføre før de andre utfører i et "non-preemption" system. Generelt vil "preemptive" systemer gjøre høyere prioriterte prosesser mer aktive, slik at de blir foretrukket. En alternativ strategi mellom "preemption" og "non-preemption" er å la lavere prioriterte prosesser fortsette å bli utført for et avgrenset/begrenset tidsintervall (ikke nødvendigvis fullføring). Disse systemene er kjent som "deferred preemption" eller "cooperative dispatching" [36].

"Preemption" i databehandling betyr midlertidig avbryting av en oppgave som utføres av et datasystem. Dette skjer uavhengig av oppgavens fullføringsgrad, og med den hensikt å gjenoppta oppgaven på et senere tidspunkt. En slik endring er kjent som en "context switch"<sup>5</sup>. Det utføres vanligvis av en privilegert oppgave eller deler av et system som kalles en preemptive "scheduler", som er i stand til å avbryte, og senere gjenoppta andre oppgaver i systemet [30].

### 2.10.1 "Preemptive multitasking"

Begrepet "preemptive multitasking" brukes til å skille et multitasking operativsystem, som tillater "preemption" av oppgaver, fra et samarbeidende multitasking system der prosesser eller oppgaver må eksplisitt programmeres til å frigjøre systemressurser når de ikke trenger.

Enkelt sagt: "Preemptive" multitasking bruker avbruddsmekanisme hvor den suspenderer gjeldende kjøreprosess og påkaller en "scheduler" for å determinere hvilken prosess som bør utføres neste gang. Derfor vil alle prosesser få noe andel av CPU-tiden til enhver tid [30].

Når en bruker får en linje i et linjesvitsjingsystem, vil kommunikasjonsbåndbredden av linjen være dedikert, selv om det ikke brukes. Når systemet er saturert, vil forespørsel om konfigurering av ekstra linjer være blokkert. For at brukere med høy prioritering skal få adgang til linjeressurser, blir systemer som "AUTOVON"<sup>6</sup> benyttet for å kombinere en prioritet til hvert

---

<sup>5</sup> En context switch er en datamaskinprosess for lagring og gjenoppretting av tilstand av CPU slik at flere prosesser kan dele en enkelt CPU ressurs [30].

<sup>6</sup> AUTOVON (Automatisk Voice Network), var et amerikansk militærtelefonsystem som ble bygget i 1963 for å

telefonapparat. Apparater med høy prioritering kan avbryte linjeressurser slik at lavere prioriterte brukere kuttes ut [31].

I et meldingssystem som "AUTODIN"<sup>7</sup> innkommende trafikk er lagret på disk (eller magnettrommel eller tape) og behandles i rekkefølge av prioritering. Når en høyre prioritet melding er lagt inn i systemet, blir meldingen behandlet og sendt så raskt som mulig. En høy prioriteringsmelding som ankommer reisemålet (meldings-switch), kan avbryte bruk av "output" enheter på bryteren, slik at utskrift av en lavere prioritet meldingen blir avbrutt [31].

I pakkesvitsjingsystemer er det liten eller ingen lagring for transportnettet [43], slik at prioritering har liten innvirkning på forsinkelsen for behandlingen av en pakke. Men når et pakkesvitsjingsystem når metning, avviser det mer trafikktilbud. Prioritering kan brukes i pakkesvitsjingsystemer til å sortere og prioritere adgang til systemet [31]. Prioritering er generelt et verktøy for å avgjøre om hvordan ressursene fordeles når systemene er mettet [44].

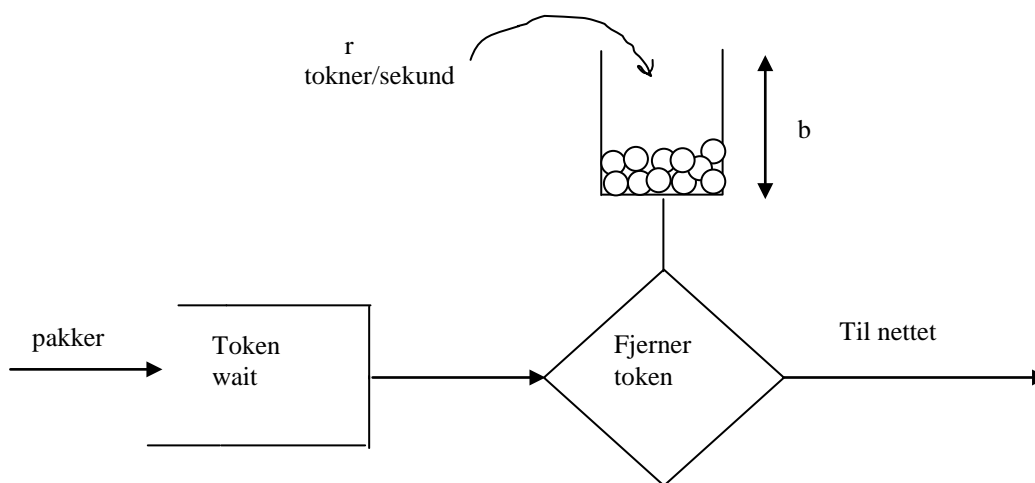
---

overleve kjernefysisk angrep [32].

<sup>7</sup> "Automatiske Digital Network (AUTODIN)" er en arv datakommunikasjon tjeneste i "United States Department of Defense".

## 2.11 "Token Bucket" modell

"Token Bucket" model som er brukt for "non-preemption" er en policy i denne oppgaven. "Token Bucket" representerer en policy av trafikkbehandling i "Diffserv". Figur 2-6 viser en "token bucket" strømning som er definert ved  $r$  og  $b$ , hvor  $r$  betegner en rate som tokner er akkumulert og  $b$  er dybden på "tokenspool" (i byter) [19].



Figur.2-6 ILLUSTRERER "TOKEN BUCKET" MODELL [19]

Som det er vist over, legges nye "token" til "bucket" med rate  $r$  (tokner/sekund). Maksimum "token" som akkumuleres er  $b$  byter. Hvis "bucket" er full, vil innkommende tokner bli kastet. "Token Bucket (TB)" profil inneholder tre parametre: en gjennomsnittlig rate, en topp rate og en "burst" rate. Vi skal se på betydningen av disse parameterne [19][20]:

### 2.11.1 Gjennomsnittlig rate

Nettet kan vurderes til å begrense langsiktig gjennomsnittlige rate slik at strømningpakker sendes inn i nettet. Det viktigste poenget her er tidsintervallet over gjennomsnittlig rate som skal overvåkes. For eksempel, en strømning som har 10000 pakker/sekund, er mer fleksibel enn en strømning som har 10 pakker/millisekund. Selv om begge strømninger har samme gjennomsnittlige rate, er deres oppførsel forskjellige. Strømning som har 10000 pakker/sekund, kan sende 100 pakker i én millisekund på lengre sikt, men strømning som har 10 pakker/millisekund kan ikke sende 100 pakker i én millisekund på lengre sikt. Dette er en viktig karakteristisk egenskap man bør vurdere [19].

"Peak rate": Definerer et maksimum rate på pakker som kan sendes i et kort tidsintervall. I eksemplet ovenfor, selv om strømningens gjennomsnittsrate er 10000 pakker/sekund, kan "peak rate" begrenses til 200 pakker/millisekund [19][20].

### 2.11.2 "Burst size"

"Burst size" er maksimal antall pakker som kan overføres på et kort tidsintervall. Med andre ord, når tidsintervallet tilnærmes null, vil "burst size" begrense antall pakker som kan overføres til nettet. Når man ikke vurderer tidsintervallet som null, vil "burst size" være avhengig av "bucketsdybde"  $b$  og maksimum hastigheten av utdata [20].

## 2.12 "Token Bucket i Differentiated Services"

Operasjonen av "Token Bucket" i diffserv kan defineres som følgende [19][65]:

- Ankommende pakker av  $r$  byter behandles umiddelbart dersom det er minst  $r$  tokner i "bucket" (én "token" = én byte) [19].
- Hvis gjeldende antall akkumulerte tokner  $b$  er mindre enn antall ankommende pakker  $r$ ,  $r-b$  pakker er feilaktige.
- Pakker er tillatt for en gjennomsnittlig rate i serieopptak opp til "burst size", så lenge de ikke overstiger topprate slik at "bucket" er tømt.
- Hvis det ikke er noen pakker som skal overføres, kan tokner bli akkumulert opp til en størrelse av  $b$ . Resten av tokner vil bli kastet.

Tilpassede og ikke-tilpassede funksjoner er avhengige av produsent/forbruker. Hvis pakkene ikke er tilpasset, kan følgende tiltak iverksettes [19][65].

- Pakken kan bli kastet.
- Pakken kan merkes på nytt på en spesiell måte.
- Pakken kan bli bufret (ved å sette inn en buffer mellom innstrømming og bestemmelsesmål) før tilstrekkelig antall tokner ankommer inn i "bucket".

Disse funksjonene brukes til forskjellige klasser på forskjellige måter [65].



---

## 2.13 Tjenestekvalitet (QoS)

Tjenestekvalitet eller "Quality of Service (QoS)" er et av de begrepene som brukes mest i datakommunikasjon i dag. De benyttes veldig hyppig, men skaper også en del forvirring. Forklaringen kan være at det finnes så mange ulike definisjoner av emnet. Forbrukere, forskere, forskjellige bransjer og fagfolk har sin egen definisjon av hva QoS er, hva QoS vil gjøre og hvordan en effektivt kan levere det [4]. Disse ulike definisjonene kan ofte føre til konflikter som involverer komplekse, upraktiske og ukompatible mekanismer for å levere de ønskede resultatene. For at en skal være i stand til å kunne levere tjenestekvalitet, er man først nødt til å forstå og definere problemet. Analyse av begrepet QoS vil sjelden føre til en enkel definisjon, delvis fordi at selve begrepet QoS er et tvetydig begrep. Klassifiseringen av pakkene, bestemmer QoS klassen for hver pakke. For hvert utgående grensesnitt, gir pakkeplanleggeren den lovede tjenestekvaliteten. Flere definisjoner av et begrep vil resultere i flere mulige løsninger [6] [5].

### 2.13.1 Hva er Quality of Service?

For å kunne forstå begrepet "Quality of Service", er en først nødt til å forstå ordene "*quality*" og "*service*".

"*Quality*" kan generelt brukes for å beskrive prosessen med å levere data på en pålitelig og kanskje bedre måte enn normalt. Definisjonen omfatter temaer som tap av data, minimal eller ingen forsinkelse, konstante forsinkelseskarakteristikkene "jitter" og muligheten til å bestemme den mest effektive måten å utnytte nettverksressursene på (f.eks. den korteste veien mellom to kommuniserende noder). "*Quality*" kan også bety en spesiell egenskap, noe som gjør at "*quality*" også blir brukt til å definere spesielle karakteristikkene ved bestemte nettverksapplikasjoner eller protokoller [3] [7].

### 2.14 Ytelse:

I likhet med pålitelighet er ytelse et begrep som vi kjenner fra dagligtalen. Innen kommunikasjonssystem så benyttes ofte termen trafikkevne i stedet for ytelse. Ytelsen kan defineres som: Mulighetene og evnen et system har til å stille ressurser til rådighet for å utføre de tjenestene det skal levere. Med andre ord, ytelsen er en konsekvens av et endelig antall ressurser i et system og den bruken systemet gjør av ressursene. I dette tilfelle kan man se på ressurstildeling av "diffServ" [50].

## 2.15 Synkronisering

Introdusering av grunnleggende synkronisering ofte kalles produsent/forbruker. Man kan forestille seg to samarbeidende enheter, hvor den ene er produsent mens den andre er forbruker. Typiske kode- skjeletter for produsent og forbruker er [66]:

```
"producer" = "make item; give item; repeat";
```

```
"consumer" = "take item; use item; repeat";
```

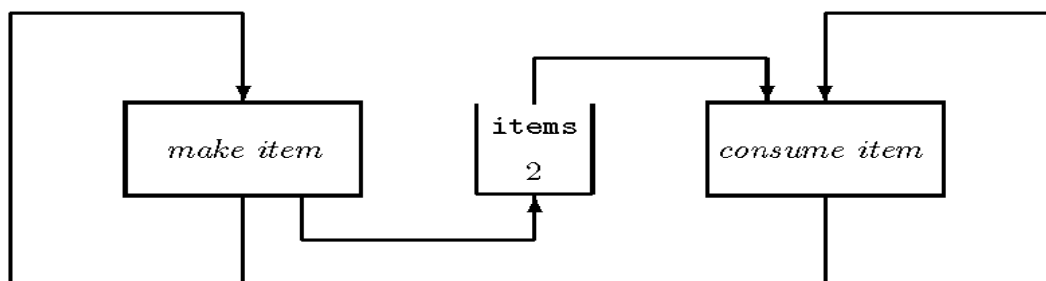
Poenget med dette er at forbrukeren er blokkert dersom ingen dataelementer er tilgjengelige for øyeblikket. Med andre ord blir dataelementer raskere forbrukt enn de blir produsert [66].

I Demos, representeres en slik pool av tilgjengelige elementer ved et "bin" objekt (se figur 2-7). For dette eksempelet, kan man lage en start pool av to elementer ved:

```
items :- new bin("items",2);          [ref(bin) items;]
```

Parameteren "text title" (sendes som "items") brukes i rapporter og i "traces", men parameteren "integer avail" gir startstørrelsen på poolen (her er 2, som vist i figur 5-1). Deretter, på samme måte som "class res", vil "avail" opprettholdes for å registrere nåværende nivå på tilgjengelige pool. Merk imidlertid at "avail" har ingen øvre grense ( $0 \leq \text{avail}$ ).

Forbrukeren får et element fra pool ved "items.take (1)". Hvis poolen er tomt (avail = 0), må forbrukeren vente i en skjult kø, q, lokalt til "items". Når det er gitt tillatelse, reduserer forbrukeren poolstørrelsen av mengden som er bedt om. Forbrukeren går deretter ut av køen og inn i en hendelsesliste ved gjeldende klokke, men bak nåværende produsent. Når produsenten har avsluttet et element, vil den oppdatere poolstørrelsen ved "items.give" (1). Denne kommandoen altså vekker forbrukeren som bør være blokkert, og gjør at den kan fortsette nå (ved gjeldende klokke, men bak nåværende produsent). Produsenten vil deretter starte på neste element [66].

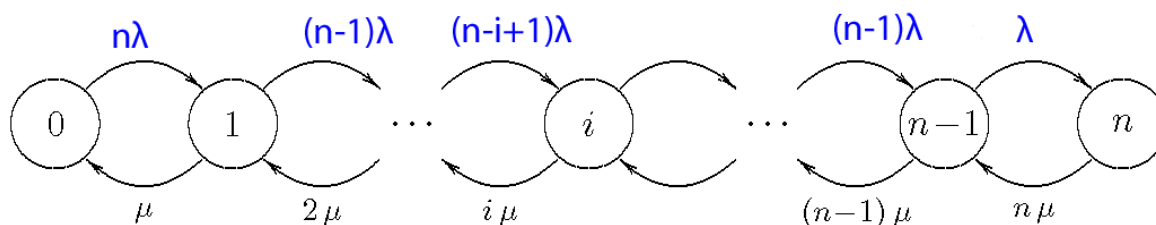


Figur.2-7 PRODUSENT/FORBRUKER AKTIVITETSDIAGRAM [66]

### 3 Trafikk modell for tale, video og data

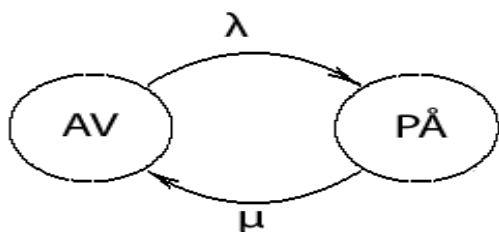
En rekke av tale-, video- og datamodeller er representert ved AV/PÅ -kildemodeller. Vi kan bruke disse enkle og fleksible modellene for trafikklider. Når en kilde er PÅ, genereres pakker med et konstant ankomsttidsintervall, mens det genereres ingen pakker når kilden er AV (se figur 3-9). Uansett tale-, video- og datakildemodellering, kan prosessen med et datakall som passerer mellom AV/PÅ-tilstander, modelleres som en to-tilstands Markovkjede [71].

Varigheten av hver tilstand er en negativ eksponential fordeling og tilstandsendingen i hver tilstand genereres av en Poissonprosess. Hvis det da til sammen er  $n$  talekilder, er det tilstrekkelig å holde rede på antall  $i$  som til enhver tid er i talefase ettersom de i prosessene til sammen virker som en Poissonprosess med intensitet  $i\mu$ . Samtidig vil de  $(n - i)$  prosessene som er i lyttefase virke sammen som en Poissonprosess med intensitet  $(n - i)\lambda$  (se figur 3-8)[72].



**Figur 3-8 TILSTANDSDIAGRAM FOR ET SYSTEM MED N TALEKILDER**

Tilstandsovergangsdigram vist i figur 3-9 viser hvordan tilstandsovergang skjer på en slik måte at tidsbrukt på hver tilstand er eksponentielt fordelt, og gitt den nåværende tilstanden av kildetraffikk, fremtiden er uavhengig av fortiden (Markovkjede).  $\lambda$  er AV- til PÅ-rate og  $\mu$  er PÅ- til AV-rate og gjennomsnittlig lengde av AV/PÅ perioder er  $1/\mu$  og  $1/\lambda$  hhv. [71].



**Figur 3-9 SYSTEM MED TO TILSTANDER**

For alle sanntidstjenester, bør tale bli generert i henhold til en Poissonprosess som forutsetter en gjennomsnittlig samtalevarighet på 120 sekunder for tale og linjeswitched data tjenester [71].

Som nevnt ovenfor, trafikkmodell for tale bør være en AV/PÅ -modell med aktive og stille perioder som genereres av en eksponentiell fordeling med middelværdier for aktive og stille perioder. Begge to skal være lik 3 sekunder [71].

### **3.1 Simulator modell**

I innledningskapitlet (kapitel 1) ble det nevnt at trafikkilder (tale, video og data) skal genereres over en "DiffServ" (se skjematisk oversikt på figur 3-10 og 3-11). I dette avsnittet kan man se litt nærmere på hva simulering er og hvordan man kan bruke dette til metodisk analyse [72]. For en generell oversikt over ulike steg i et simuleringsstudium, henvises det til boka "Pålitelighet og ytelse i informasjons- og kommunikasjonssystem grunnlag" som står i litteraturlista med nummer 72.

Simulering er en etterligning av et konstruert hendelsesforløp ved hjelp av datamaskin. Å simulere, er det samme som å representere visse hovedegenskaper eller atferd av et valgt fysisk eller abstrakt system [74].

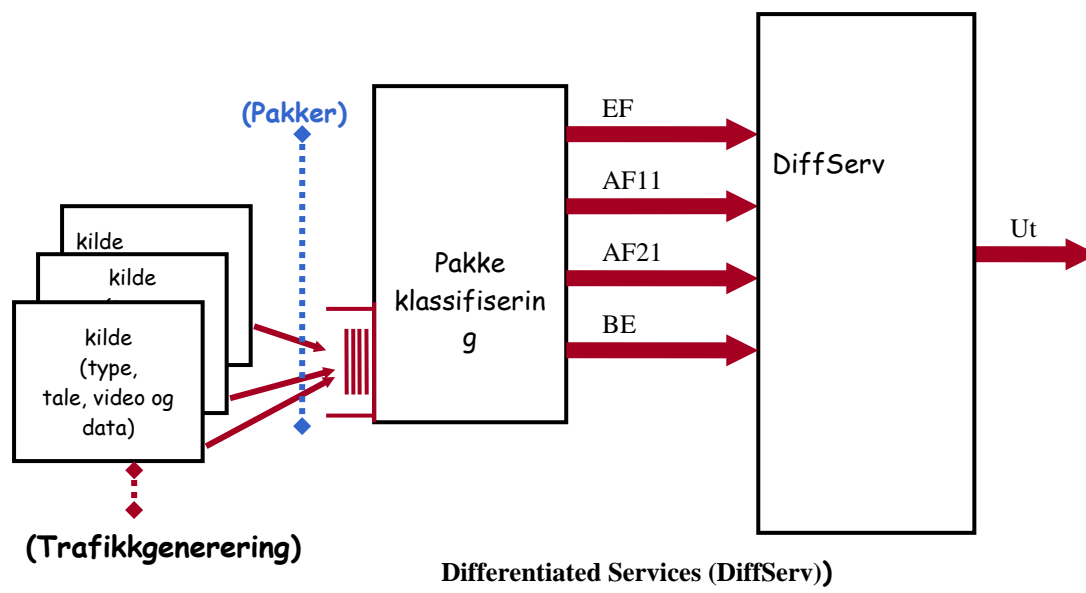
Hva gjør simulering med oss?

- Man lærer bedre det man skal gjøre
- Man lærer mer om det man skal gjøre
- Man blir tryggere på det man skal gjøre
- Man gjør mindre feil [74]

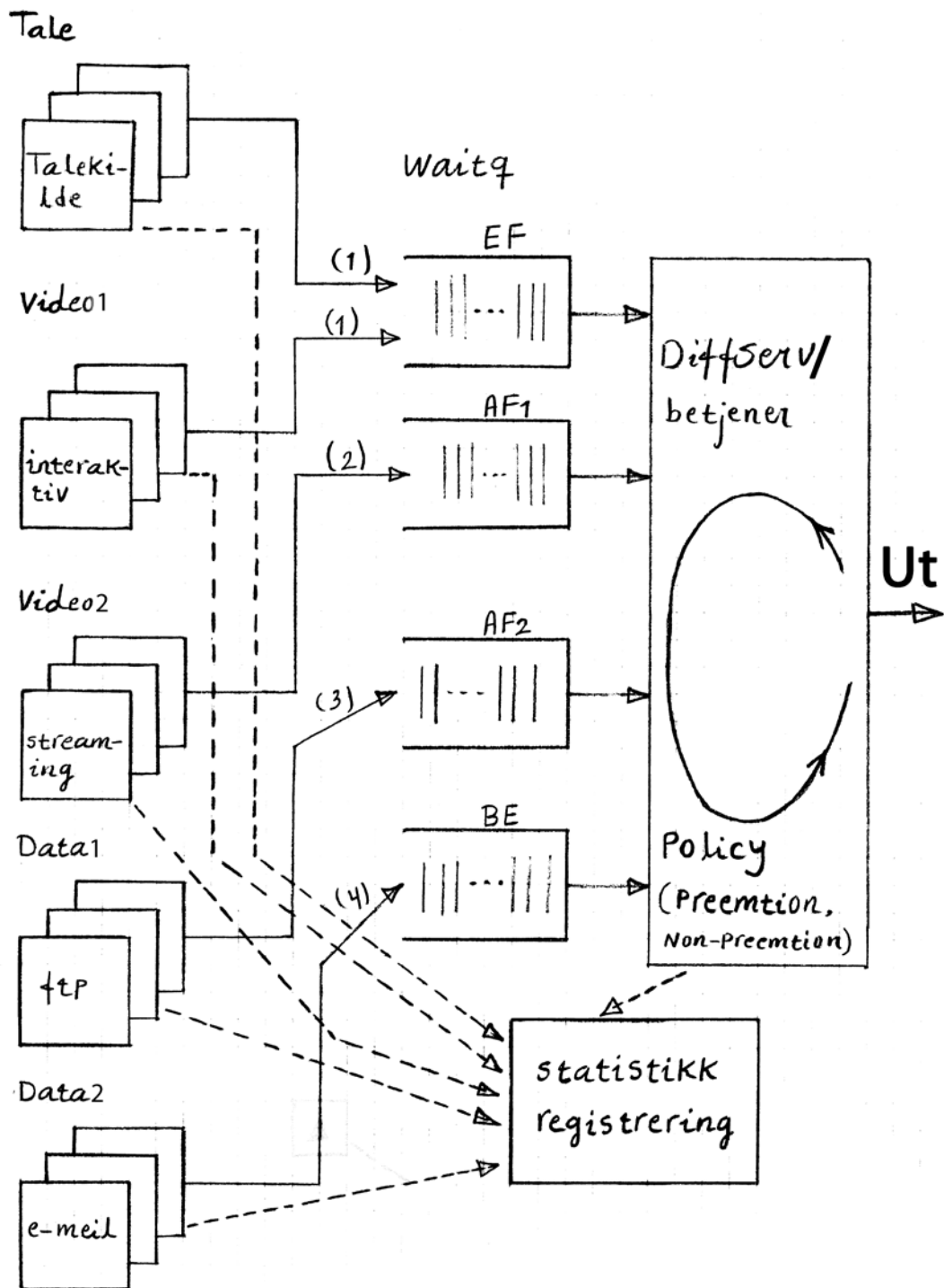
I denne simuleringsmodellen sammenligner man trafikkildene med hensyn på tap og forsinkelse. En generell beskrivelse av simulatormodellen er illustrert i figur 3-3 og figur 3-4.

Modellen består av 3 hoveddeler:

1. Kildetraffic; tale, video og data (se kapitel 4 avsnitt 4.1 – 4.2)
2. En pakkeklassifisering som er egentlig en del av "Differentiated Services (DiffServ)"
3. En betjener; denne er også en del av "Diffserv (dette er beskrevet i avsnitt 2.2 – 2.5)



Figur 3-10 SIMULATOR MODELL



Figur 3-11 FORKLARENDE ILLUSTRASJON FOR SIMULATORMODELL

### **3.2 Trafikkilder:**

Vi har i denne oppgaven valgt å bruke tre kildetyper å modellere. De tre grunnleggende kildene som brukes i denne oppgaven er som følgende:

- Tale (se avsnitt 4.1 og 4.4.1)
- Video (se avsnitt 2.9)
- Data (se avsnitt 4.3.1 og 4.3.2)

## 4 Generering av tale-, video- og datakilder

### 4.1 Matematiske ligninger for tale- og videokilder

Den "kumulative fordelingsfunksjonen"<sup>8</sup> av talespurtsvarighet er avrundet ved en eksponentiell funksjon, og for pausevarigheter er den avrundet ved en "constant-plus-exsponensial" [67]. Ligning (1) er for "talk-spurt" og ligning (2) for pause [67]:

$$P_{ts}(t) = 1 - \exp(-A_{ts} * t) \quad (1)$$

$$A_{ts} = 1/\bar{T}_{ts}, \quad \bar{T}_{ts}: \text{Gjennomsnittlig av talespurtsvarighet,}$$

$$P_{ps}(t) = \begin{cases} 0 & \text{for } 0 \leq t \leq 0.2 \\ 1 - \exp[-A_{ps}(t-0.2)] & \text{for } t > 0.2 \end{cases} \quad (2)$$

$$A_{ps} = 1/(\bar{T}_{ps} - 0.2), \quad \bar{T}_{ps}: \text{average pause duration.}$$

I denne oppgaven er følgende ligning blitt brukt:

$$P_{ts}(t) = 1 - \exp(A_{ps} * t)$$

Ligning (3) kan brukes for en tilfeldig "talk-spurt" ved tiden t:

$$P_i(t) = 1 - \exp\left[-(n-i) \cdot \frac{1}{\lambda} + i \cdot \frac{1}{\mu}\right] \cdot t \quad (3)$$

Systemet går opp:

<sup>8</sup> Kumulative fordelingsfunksjonen beskriver en sannsynlighetsfordeling for en stokastisk variabel innenfor matematisk statistikk. For en stokastisk variabel X, med sannsynlighetsfunksjon P(x), defineres den kumulative fordelingsfunksjonen F<sub>X</sub>(x) som følgende ligning [78]:

$$F_X(x) = P(X \leq x)$$



$$P_{i,opp} = \frac{(n-i) \cdot \frac{1}{\lambda}}{(n-i) \frac{1}{\lambda} + i \cdot \frac{1}{\mu}}$$

Systemet går ned:

$$P_{i,ned} = 1 - \frac{(n-i) \cdot \frac{1}{\lambda}}{(n-i) \frac{1}{\lambda} + i \cdot \frac{1}{\mu}}$$

Systemet befinner seg i tilstand 0 ved tiden  $t = 0$ . Da er  $P_0(0) = 0$

#### 4.2 Metode for generering av kunstige tale [67]

"Talk-spurt" og "pause" er generert i henhold til tilstandsovergangsmodellen som er vist i figur 4-12, der  $P_1$ ,  $P_2$  og  $P_3$  betegner overgangssannsynligheter som er uttrykt i prosenter.

$T_{st}$  (single talk duration),  $T_{dt}$  (double talk duration), og  $T_{ms}$  (mutual silence duration) varierer i henhold til følgende likninger. Tidsenhet i disse ligningene er uttrykt i sekunder [67].

$$T_{st} = -0.854 \ln(1-x_1)$$

$$T_{dt} = -0.226 \ln(1-x_2)$$

$$T_{ms} = -0.456 \ln(1-x_3)$$

$$0 < x_1, x_2, x_3 < 1$$

Følgende beregninger er basert på tilfeldige variabler ved normalfordeling:

$$x_1 = 0.7$$

$$T_{st} = -0.858 \ln(1-0.7) = 1.02819$$

$$x_1 = 0.6999$$

$$T_{st} = -0.858 \ln(1-0.6999) = 1.02791$$

$$x_1 = 0.6$$

$$T_{st} = -0.858 \ln(1-0.6) = 0.782512$$

$$x_2 = 0.98829$$

$$T_{dt} = -0.226 \ln(1-0.98829) = 1.0049$$

$$x_2 = 0.7$$

$$T_{dt} = -0.226 \ln(1-0.7) = 0.272098$$

$$x_2 = 0.6999$$

$$T_{dt} = -0.226 \ln(1-0.6999) = 0.272023$$

$$x_2 = 0.6$$

$$T_{dt} = -0.226 \ln(1-0.6) = 0.207082$$

$$x_2 = 0.4$$

$$T_{dt} = -0.226 \ln(1-0.4) = 0.115447$$

$$x_3 = 0.9$$

$$T_{ms} = -0.456 \ln(1-0.9) = 1.004998$$

$$x_3 = 0.8895$$

$$T_{ms} = -0.456 \ln(1-0.8895) = 1.00445$$

$$x_3 = 0.7$$

$$T_{ms} = -0.456 \ln(1-0.7) = 0.549012$$

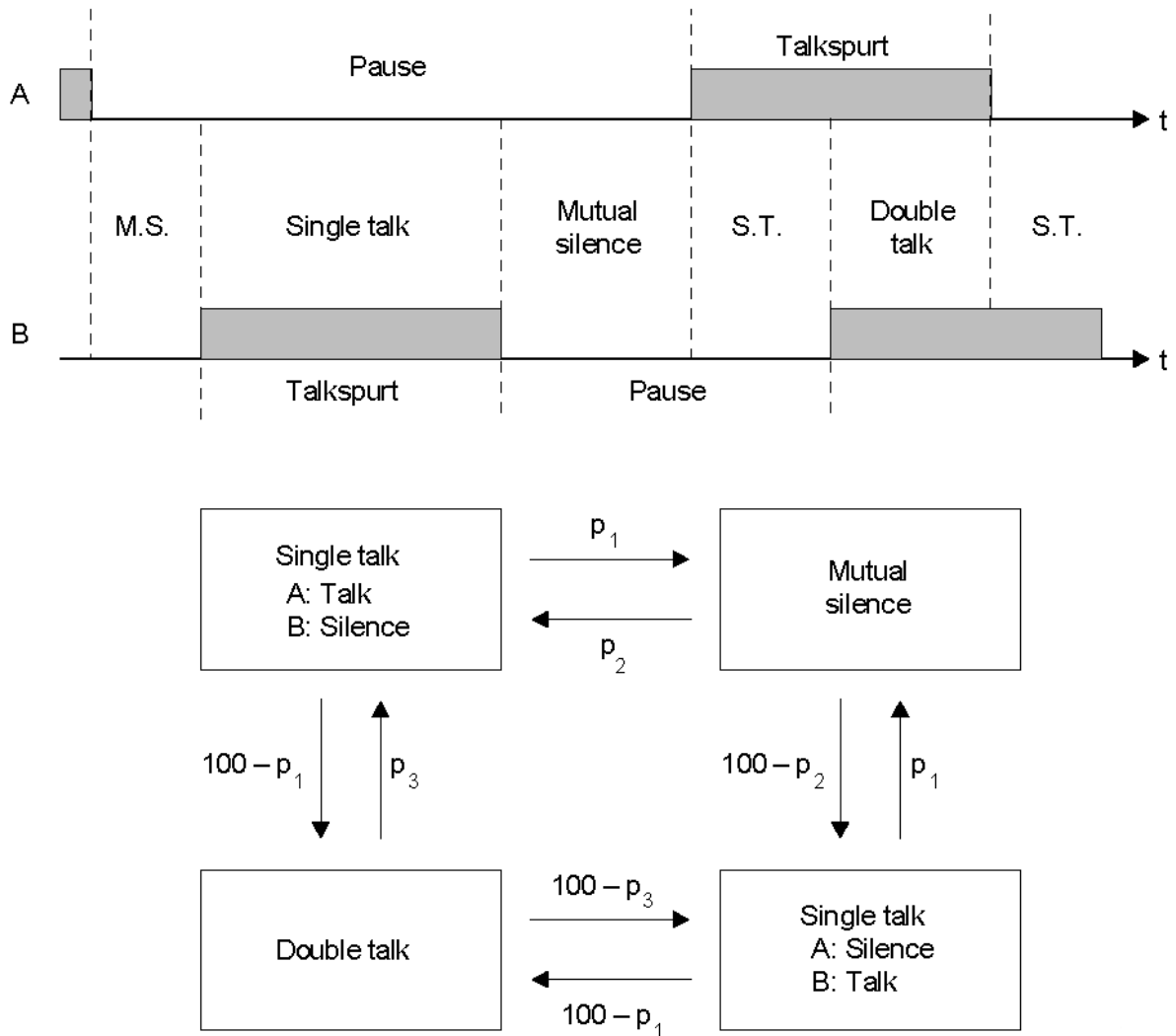
$$x_3 = 0.6$$

$$T_{ms} = -0.456 \ln(1-0.6) = 0.417829$$

$$x_3 = 0.4$$

$$T_{ms} = -0.456 \ln(1-0.4) = 0.232936$$

Fra beregningene ovenfor ser man at med høyere verdier for  $x$ , blir høyere tidsenhet for både  $T_{st}$  (single talk duration),  $T_{dt}$  (double talk duration), og  $T_{ms}$  (mutual silence duration) [67].



Figur 4-12 TILSTANDSOVERGANGSMODELL FOR SAMTALE [67]

Tabell 4-2 Gjennomsnittlig perioder for AV/PÅ talekilder [67].

Parameter	Rate (s)	Rate (%)
$\mu$	1.004	38.53
$\Lambda$	1.587	61.47
Data rate 64 kbps		

---

### 4.3 Beregning av AV/PÅ varighet for e-post og ftp

Beregning av AV/PÅ for trafikktypene e-post (BE-klasse) og FTP (AF2-klasse) er basert på tabell 4-3, 4-4 og 4-6. Datarate er 384 kilobyte per sekund.

#### 4.3.1 E-post

En beregner først e-post trafikk med hensyn til tredje kolonne i tabell 4-4. Det er 90 sekunder intervall ankomsttid mellom pakkekaller, 25 pakker per pakkekall, 480 bytes gjennomsnittlig datarate pr. kilde og 0,067 sekunder intervall ankomsttid mellom pakkene. Ifølge tabell 4-3 har vi 5 pakkekall per sesjon [68].

Gjennomsnittlig tid for en pakke er  $480 \times 8 / 384000 = 0,01$  sekunder og AV-tiden mellom to etterfølgende pakker er  $0,067 - 0,01 = 0,057$  sekunder og AV-tiden i en pakkekall  $(15-1) \times 0,057 = 0,8$  sekunder. Hele AV-tiden mellom pakkekaller er  $1 \times 0,8 = 0,8$  sekunder. Den totale AV-tiden i en sesjon er  $1 \times 90 + 0,8 = 90,8$  sekunder. Deretter kan man beregne PÅ-tiden av en e-post sesjon i en pakkekall som er  $15 \times 0,01 = 0,15$  sekunder, slik at den totale PÅ-tiden i en e-post sesjon blir  $2 \times 0,15 = 0,3$  sekunder [68].

#### 4.3.2 ftp

For ftp er null sekunder intervall ankomsttid mellom pakkekaller, 25 pakker per pakkekall, 480 bytes gjennomsnittlig datarate pr. kilde og 0,067 sekunder intervall ankomsttid mellom pakkene. Gjennomsnittlig tid for en pakke er  $480 \times 8 / 384000 = 0,01$  sekunder, AV-tiden mellom to etterfølgende pakker er  $0,067 - 0,01 = 0,057$  sekunder og AV-tiden i en pakkekall  $(62-1) \times 0,057 = 3,477$  sekunder. Den totale AV-tiden mellom pakkekaller er  $1 \times 3,477 = 3,477$  sekunder. Total AV-tid i en sesjon er  $1 \times 0 + 3,477 = 3,477$  sekunder. En kan deretter beregne PÅ-tiden av en e-post sesjon i en pakkekall som er  $62 \times 0,01 = 0,62$  sekunder, slik at total PÅ-tid i en e-post sesjon blir  $1 \times 0,62 = 0,62$  sekunder [68].

**Tabell 4-3 Forkjellige type informasjon for datagram [68]**

Packet based information types	Average number of packet calls within a session	Average reading time between packet calls [s]	Average amount of packets within a packet call []	Average interarrival time between packets [s]	Parameters for packet size distribution
WWW surfing UDD 8 kbps	5	412	25	0,5	K=81,5 $\alpha=1,1$
WWW surfing UDD 32 kbps	5	412	25	0,25	K=81,5 $\alpha=1,1$
WWW surfing UDD 64 kbps	5	412	25	0,0625	K=81,5 $\alpha=1,1$
WWW surfing UDD 144 kbps	5	412	25	0,0277	K=81,5 $\alpha=1,1$
WWW surfing UDD 384 kbps	5	412	25	0,01040	K=81,5 $\alpha=1,1$
WWW surfing UDD 2048 kbps	5	412	25	0,00195	K=81,5 $\alpha=1,1$

**Tabell 4-4 Trafikkmodellens egenskaper [68]**

Modell Parameter	ftp	e-post
No. Of Packet Calls per Session	1	Geometric (mean of 2)
Inter Arrival Time or Reading Time Between Packet Calls (sec)	-	Pareto (mean of 90)
No. Of Packets per Packet Call	Pareto (mean of 25)	Weibull distributed packet cal size (mean of 15)
Packet size (bytes)	480	480
Inter-Arrival Time Between Packets (sec)	Geometric (mean of 0,067)	Geometric (mean of 0,067)

#### 4.4 Beregning av "time spacing" for business telefoni (EF-klasse)

- Toppoverføringsrate for tale er lik 64 kilobit per sekund = 64000 bit/sekund [70]
- Gjennomsnittelig datarate pr. kilde er lik 24,8 kilobit per sekund = 24800 bit/sekund [70]
- Gjennomsnittstid "time spacing" av en pakke er lik x

$$64000 \frac{\text{bit}}{\text{sekund}} * x_{\text{sekund}} = 24800 \frac{\text{bit}}{\text{sekund}}$$

$$x_{\text{sekund}} = \frac{24800 \frac{\text{bit}}{\text{sekund}}}{64000 \frac{\text{bit}}{\text{sekund}}} 0.3875 \Rightarrow x = 0.3875$$

#### 4.4.1 Gjennomsnittlig beregning av overføringstid for tale

Antall biter per pakke multipliseres med 8 for å få 8 bit per pakke, og deles med 64000 bit per sekund for å få gjennomsnittlig overføringstid per pakke som følgende:

$(24800 \text{ bit per pakke}) * (8 \text{ bit}) = (198400 \text{ bit per pakke}) / (64000 \text{ bit per sekund}) = 3,10000$  sekunder til å overføre.

Følgende data er brukt for tale i simulatoren, disse er gjenbrukt fra "STOLAS" [69]:

- pakkerate: 8,33333
- pakkelenge: 7680

#### 4.6 Beregning av "time spacing" for videokilder (EF- og AF1-klasse) [70]:

- Toppoverføringsrate for video1(interaktiv) og video2 (streaming) er lik 945 kbps = 945000 bit/sekund.
- Gjennomsnittlig datarate pr. kilde er lik 1500 bit/sekund
- Gjennomsnittstid "time spacing" av en pakke er lik x

$$945000 \frac{\text{bit}}{\text{sekund}} * x_{\text{sekund}} = 1500 \frac{\text{bit}}{\text{sekund}}$$

$$x_{\text{sekund}} = \frac{1500 \frac{\text{bit}}{\text{sekund}}}{945000 \frac{\text{bit}}{\text{sekund}}} = 0.015873 \Rightarrow x = 0.015873_{\text{sekund}}$$

#### 4.6.1 Gjennomsnittlig beregning av overføringstid for video1 og video2 [70]

Antall biter per pakke multipliseres med 8 for å få 8 biter, deretter deles med 945000 biter per sekund for å få gjennomsnittlig overføringstid per pakke som følgende:

Forfatteren har valgt 1500 bit per pakke for både video1 og video2.

$(1500 \text{ bit per pakke}) * (8 \text{ bit}) = (12000 \text{ bit per pakke}) / (945000 \text{ bit per sekund}) = 0,012698$   
sekunder til å overføre.

#### 4.6.2 Beregning av pakkerate for video1 og video2

Pakkelenge for video kan variere mellom 1-4095 bytes (8-32760 bit) [73].

I denne simuleringen er pakkelenge på 1500 bit per pakke brukt for video1 og video 2.

- pakkelenge:  $(1500 \text{ bit per pakke}) * (8 \text{ bit}) = 12000 \text{ bit}$
- aktiv rate = 945000 bit/sekund [70]
- pakkerate =  $\frac{\text{Aktiv rate}}{\text{Pakkelenge}} = \frac{945000 \text{ bit / sekund}}{12000 \text{ bit}} = 78.75 \text{ pakker per sekund}$

**Tabell4-5 Gjennomsnittlig perioder for AV/PÅ (video1 og video2) [70]**

	PÅ	AV
Video1	0,321	1,273
Vide2	0,321	1,273
Data rate 945 kbps		

#### 4.7 Beregning av "time spacing" for data1 og data2 (AF2 og BE) henholdsvis e-mail og ftp [71]

- Toppoverføringsrate for data 1 (ftp) og data 2 (e-mail) er lik 64kbs = 64000 bit/sekund
- Gjennomsnittelig datarate pr. kilde er lik 480 bit/sekund
- Gjennomsnittstid "time spacing" av en pakke er lik x

$$64000 \frac{\text{bit}}{\text{sekund}} * x_{\text{sekund}} = 480 \frac{\text{bit}}{\text{sekund}}$$

$$x_{\text{sekund}} = \frac{480 \frac{\text{bit}}{\text{sekund}}}{64000 \frac{\text{bit}}{\text{sekund}}} = 0.06 \Rightarrow x = 0.06_{\text{sekun}}$$

#### 4.7.1 Gjennomsnittlig beregning av overføringstid for data1 og data2 [71]

Antall biter per pakke multipliseres med 8 for å få 8 biter, og deles deretter med 384000 biter per sekund for å få gjennomsnittlig overføringstid per pakke som følgende:

$(480 \text{ bit per pakke}) * (8 \text{ bit}) = (3840 \text{ bit per pakke}) / (384000 \text{ bit per sekund}) = 0,10000$  sekunder til å overføre.

#### 4.7.2 Beregning av pakkerate for data1 og data 2:

- pakkelenge:  $(480 \text{ bit per pakke}) * (8 \text{ bit}) = 3840 \text{ bit}$
- aktiv rate = 384000 bit/sekund [71]

$$\text{pakkerate} = \frac{\text{Aktiv rate}}{\text{Pakkelenge}} = \frac{384000 \text{ bit / sekund}}{3840 \text{ bit}} = 100 \text{ pakker per sekund}$$

**Tabell 4-6 Gjennomsnittlig perioder for AV/PÅ (ftp og e-mail) [68]**

	PÅ	AV
e-mail	0,3	90,8
ftp	0,61	3,477
Data rate 384 kbps		



## 5 Resultater

### 5.1 Beregning av antall pakker for trafikklider

Den totale kapasiteten er normalt 100 Mbps.

Følgende ligningen brukes for å beregne gjennomsnittelig datarate per. kilde for tale, video1, video2, data1 og data2:

$$\text{datarate pr. kilde} = \left( \frac{\lambda}{\lambda + \mu} \right) * \text{Datarate}$$

Sannsynligheten for å finne systemet i tilstand 0 avtar eksponentielt med tiden. Når tiden går mot uendelig.

$$P_0(\infty) = \frac{\frac{1}{\lambda}}{\frac{1}{\mu} + \frac{1}{\lambda}}$$

Dette er for stasjonærløsning, der er  $1/\lambda$  og  $1/\mu$  forventet tid i henholdsvis tilstand 0 og 1 [72].

#### 5.1.1 Hastighetsberegning for 90 Mbps:

90 Mbps er henholdsvis fordelt 18 Mbps for tale, 18 Mbps for video 1, 18 Mbps for video 2, 32,4 Mbps for data 1 og 3,6 Mbps for data2.

Gjennomsnittelig datarate pr. kilde for tale er 24,8 kbps [70]:

$$N(\text{antall taler}) = \frac{18000 \text{ kbps}}{24.8 \text{ kbps}} \approx 726$$

Beregning for gjennomsnittelig datarate pr. kilde og antall pakker for video 1 og video 2:

$x_1$  er lik gjennomsnittelig datarate pr. kilde for video 1 og video 2.

$$x_1 = \left( \frac{0.321}{0.321 + 1.273} \right) * 945 \text{ kbps} = 190.304 \text{ kbps}$$

Hvor middelverdiene 0.321s og 1.273s henvises til Tabell 4-5.

$$M_1 (\text{antall video 1}) = \frac{18000 \text{ kbps}}{190.304 \text{ kbps}} \approx 95$$

$$M_1 (\text{antall video 2}) = \frac{18000 \text{ kbps}}{190.304 \text{ kbps}} \approx 95$$

Beregning for gjennomsnittelig datarate pr. kilde og antall pakker for data 1 og data 2:  
 $x_1$  er lik gjennomsnittelig datarate pr. kilde for data 1.

$$x_2 = \left( \frac{0.61}{0.61 + 3.477} \right) * 384 \text{ kbps} = 57.3134 \text{ kbps}$$

Hvor middelverdiene 0.61s og 3.477s henvises til Tabell 4-6.

32,4 Mbps av 90 Mbps brukes for data1(ftp):

$$D_1 (\text{antall ftp}) = \frac{32400 \text{ kbps}}{57.3134 \text{ kbps}} \approx 565$$

$x_3$  er lik gjennomsnittelig datarate pr. kilde for data 1:

$$x_3 = \left( \frac{0.3}{0.3 + 90.8} \right) * 384 \text{ kbps} = 1.26454 \text{ kbps}$$

Hvor middelverdiene 0.3s og 90.8s henvises til Tabell 4-6.

Og 3,6 Mbps av 90 Mbps brukes for data 2 (e-mail):

$$D_2 (\text{antall e\_mail}) = \frac{3600 \text{ kbps}}{1.26454 \text{ kbps}} \approx 2847$$

Ligning for total hastighetskapasitet som skal brukes:

$$\begin{aligned} TOT = N * (\text{pakke størrelse for tale}) + (M_1 + M_2) \\ * (\text{pakke størrelse for video 1 og video 2}) + D_1 \\ * (\text{pakke størrelse for ftp}) + D_2 * (\text{pakke størrelse for e\_mail}) \end{aligned}$$

Beregning av total hastighet for 90 prosent:

$$TOT = 726 * 24.8 \text{ kbps} + (95 + 95) * 190.304 \text{ kbps} + 565 * 57.3134 \text{ kbps} + 2847 \\ * 1.26454 \text{ kbps} = 90144.9 \text{ kbps}$$

Det er ca. 90,1449 Mbps som er litt større enn 90 Mbps som er blitt brukt.

### 5.1.2 Hastighetsberegning for 80 Mbps:

80 Mbps er henholdsvis fordelt 16 Mbps for tale, 16 Mbps for video 1, 16 Mbps for video 2, 28,8 Mbps for data 1 og 3,2 Mbps for data2.

Gjennomsnittelig datarate pr. kilde for tale er 24,8 kbps:

$$N (\text{antall taler}) = \frac{16000 \text{ kbps}}{24.8 \text{ kbps}} \approx 645$$

Beregning for gjennomsnittelig datarate pr. kilde og antall pakker for video 1 og video 2:

$x_1$  er lik gjennomsnittelig datarate pr. kilde for video 1 og video 2.

$$x_1 = \left( \frac{0.321}{0.321 + 1.273} \right) * 945 \text{ kbps} = 190.304 \text{ kbps}$$

$$M_1 (\text{antall video 1}) = \frac{16000 \text{ kbps}}{190.304 \text{ kbps}} \approx 84$$

$$M_1 (\text{antall video 2}) = \frac{16000 \text{ kbps}}{190.304 \text{ kbps}} \approx 84$$

Beregning for gjennomsnittelig datarate pr. kilde og antall pakker for data 1 og data 2:

$x_1$  er lik gjennomsnittelig datarate pr. kilde for data 1.

$$x_2 = \left( \frac{0.61}{0.61 + 3.477} \right) * 384 \text{ kbps} = 57.3134 \text{ kbps}$$

28,8 Mbps av 80 Mbps brukes for ftp:

$$D_1 (\text{antall ftp}) = \frac{28800 \text{ kbps}}{57.3134 \text{ kbps}} \approx 502$$

$x_3$  er lik gjennomsnittelig datarate pr. kilde for data 1 (ftp):

$$x_3 = \left( \frac{0.3}{0.3 + 90.8} \right) * 384 \text{ kbps} = 1.26454 \text{ kbps}$$

Og 3,2 Mbps av 80 Mbps brukes for data 2 (e-mail):

$$D_2 (\text{antall e\_mail}) = \frac{3200 \text{ kbps}}{1.26454 \text{ kbps}} \approx 2531$$

Ligning for total hastighetskapasitet som skal brukes:

$$\begin{aligned} TOT = N * (\text{pakke størrelse for tale}) + (M_1 + M_2) \\ * (\text{pakke størrelse for video 1 og video 2}) + D_1 \\ * (\text{pakke størrelse for ftp}) + D_2 * (\text{pakke størrelse for e\_mail}) \end{aligned}$$

Beregning av total hastighet for 80 prosent:

$$\begin{aligned} TOT = 645 * 24.8 \text{ kbps} + (84 + 84) * 190.304 \text{ kbps} + 502 * 57.3134 \text{ kbps} + 2531 \\ * 1.26454 \text{ kbps} = 79939 \text{ kbps} \end{aligned}$$

Det er ca. 79,939 Mbps som er litt mindre enn 80 Mbps som er blitt brukt.

### 5.1.3 Hastighetsberegning for 95 Mbps:

95 Mbps er henholdsvis fordelt 19 Mbps for tale, 19 Mbps for video 1, 19 Mbps for video 2, 34,2 Mbps for data 1 og 3,8 Mbps for data2.

Gjennomsnittelig datarate pr. kilde for tale er 24,8 kbps:

$$N (\text{antall taler}) = \frac{19000 \text{ kbps}}{24.8 \text{ kbps}} \approx 766$$

Beregning for gjennomsnittelig datarate pr. kilde og antall pakker for video 1 og video 2:

$x_1$  er lik gjennomsnittelig datarate pr. kilde for video 1 og video 2.

$$x_1 = \left( \frac{0.321}{0.321 + 1.273} \right) * 945 \text{ kbps} = 190.304 \text{ kbps}$$

$$M_1 (\text{antall video 1}) = \frac{19000 \text{ kbps}}{190.304 \text{ kbps}} \approx 100$$

$$M_1 (\text{antall video 2}) = \frac{19000 \text{ kbps}}{190.304 \text{ kbps}} \approx 100$$

Beregning for gjennomsnittelig datarate pr. kilde og antall pakker for data 1 og data 2:

$x_1$  er lik gjennomsnittelig datarate pr. kilde for data 1.

$$x_2 = \left( \frac{0.61}{0.61 + 3.477} \right) * 384 \text{ kbps} = 57.3134 \text{ kbps}$$

34,2 Mbps av 95 Mbps brukes for ftp:

$$D_1 (\text{antall ftp}) = \frac{34200 \text{ kbps}}{57.3134 \text{ kbps}} \approx 597$$

$x_3$  er lik gjennomsnittelig datarate pr. kilde for data 1 (ftp):

$$x_3 = \left( \frac{0.3}{0.3 + 90.8} \right) * 384 \text{ kbps} = 1.26454 \text{ kbps}$$

Og 3,8 Mbps av 95 Mbps brukes for data 2 (e-mail):

$$D_2 (\text{antall e-mail}) = \frac{3800 \text{ kbps}}{1.26454 \text{ kbps}} \approx 3005$$

Ligning for total hastighetskapasitet som skal brukes:

$$\begin{aligned}
 TOT &= N * (\text{pakke størrelse for tale}) + (M_1 + M_2) \\
 &\quad * (\text{pakke størrelse for video 1 og video 2}) + D_1 \\
 &\quad * (\text{pakke størrelse for ftp}) + D_2 * (\text{pakke størrelse for e\_mail})
 \end{aligned}$$

Beregning av total hastighet for 95 prosent:

$$\begin{aligned}
 TOT &= 766 * 24.8 \text{ kbps} + (100 + 100) * 190.304 \text{ kbps} + 597 * 57.3134 \text{ kbps} + 3005 \\
 &\quad * 1.26454 \text{ kbps} = 95073.6 \text{ kbps}
 \end{aligned}$$

Det er ca. 95,736 Mbps som er litt større enn 95 Mbps som er blitt brukt

## 5.2 Beregning av middelværdi, varians og konfidensintervall for trafikktyper:

Replikasjonsmetode<sup>9</sup> er blitt brukt i denne oppgaven for å samle inn 10 uavhengige observasjoner for forskjellige seed-verdier. Samme start og stoppbetingelser er benyttet i programmet.

Det er beregnet konfidens for hver av middelværdiene for trafikktyper (tale, video1, video2, data1 og data2) som er basert på 10 uavhengige observasjoner. Det innebærer at variansen for middelværdien av disse er funnet ut fra de ti uavhengige observasjonene, og konfidensintervallet er deretter beregnet for disse middelværdiene. Dette ble gjort for "preemptiv" og "non-preemptiv" med 80 og 90 prosent belastning, slik at man kan sammenligne kildetyperne med hensyn på forsinkelse og betjening.

Beregningsmåten som er valgt er det samme som er brukt internt i DEMOS (og i stolas postpros programmet [69]).

Følgende uttrykk er blitt brukt:

```
stdev := sqrt(abs(10.0 * m2sum - msum ** 2)/90.0);
```

```
konfint := stdev * 2.262 / sqrt(10.0);
```

hvor

sqrt(x) er kvadratroten av x

abs(x) er absoluttverdien av x

x \*\* 2 er kvadratet av x

"msum" er summen av alle 10 observasjonene.

<sup>9</sup> Replikasjonsmetode: er en metode for å samle inn n uavhengige identiske fordelte observasjoner ved å gjenta et eksperiment n ganger med samme start og stoppbetingelser [72].

"m2sum" er summen av de 10 observasjonene kvadrert (dvs. hver observasjon kvadreres og disse kvadratene summeres sammen), deretter deler man disse på 90.0 fordi det er  $N * (N-1) = 10 * 9 = 90$

hvor  $N =$  antall observasjoner

Øvre og nedre grense i konfidensintervallet finner man ved å legge til og trekke fra "konfint" som definert over.

I sannsynlighet og statistikk brukes student-t-fordeling for en sannsynlighetsfordeling som oppstår i problemer med å beregne gjennomsnitt av et normalt distribuert utvalg når utvalgsstørrelsen er liten [76].

Det er i denne oppgaven brukt student-t-fordelt med 9 frihetsgrader og 95 prosent konfidensfordelt med 2.5 prosent feilmargin i hver ende av fordelingen, det vil si verdien for 97.5 prosent som er 2.262 [77].

Alternativ for beregning av variansen:

$$\text{Middellverdi} = \frac{\text{summen av antall observasjoner}}{10}$$

$$(\text{Variansen})^2 = \frac{1}{10 - 1} \sum_1^{10} (\text{observasjon} - \text{middellverdi})^2$$

Konfidensintervallet:

$$\left[ \text{Middellverdi} - T * \frac{\sqrt{(\text{Variansen})^2}}{\sqrt{10}}, \text{Middellverdi} + T * \frac{\sqrt{(\text{Variansen})^2}}{\sqrt{10}} \right]$$

Hvor student-t-fordeling = T

Det brukes samme student-t-fordeling for 97.5 prosent som er 2.262.

Beregning for både "preemption" og "non-preemption" for 95 prosent belastning er gjort, men denne er ikke tatt med i resultater.

### 5.3 Registrerte og beregnede resultater i form av figurer og tabeller

Tabell 5-7 Måleresultater etter 10 ganger kjøring av 10 uavhengig observasjoner for "preemption" med 90 % belastning.

	Tale	Video1	Video2	Data1	Data2
1	1,07E-05	6,65E-06	0,000135	0,07092	45,548
2	1,07E-05	6,65E-06	0,000135	0,07379	36,188
3	1,07E-05	6,65E-06	0,000135	0,07094	50,315
4	1,07E-05	6,65E-06	0,000134	0,07149	33,696
5	1,08E-05	6,63E-06	0,000134	0,08122	61,07
6	1,07E-05	6,63E-06	0,000134	0,06870	29,178
7	1,07E-05	6,64E-06	0,000135	0,08303	62,498
8	1,07E-05	6,62E-06	0,000135	0,07923	63,161
9	1,06E-05	6,63E-06	0,000133	0,06286	15,452
10	1,07E-05	6,65E-06	0,000135	0,08343	57,346
SUM	0,000107	6,64E-05	0,001344	0,74561	454,452
<b>Forventning:</b>	<b>1,07E-05</b>	<b>6,64E-06</b>	<b>0,000134</b>	<b>0,074561</b>	<b>45,4452</b>

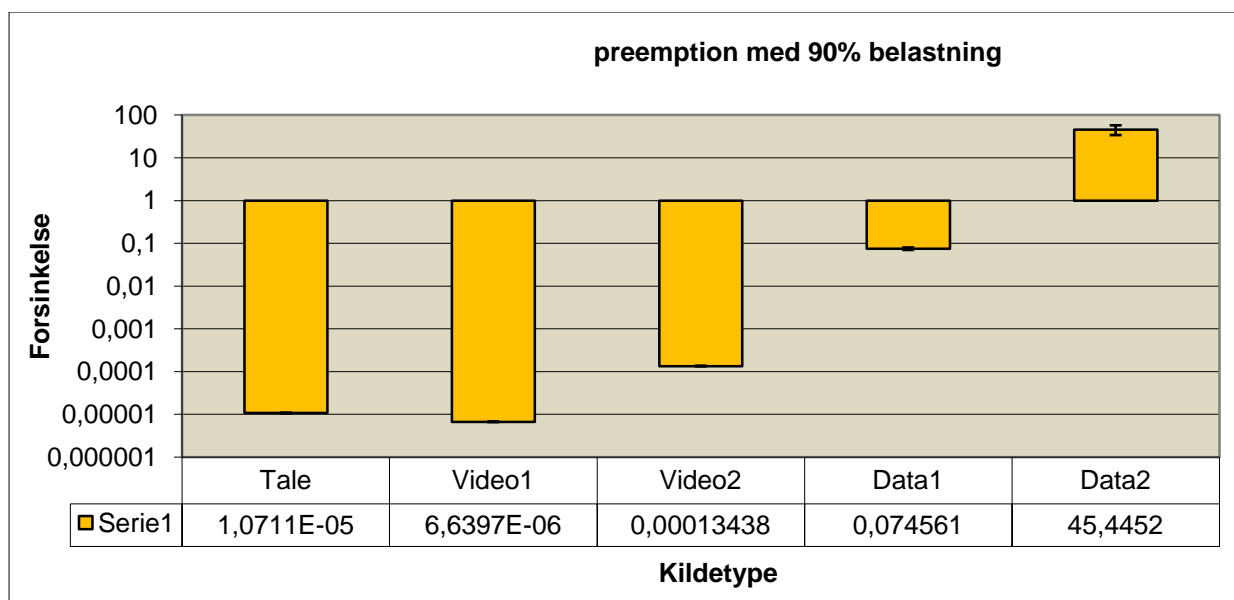
Tabell 5-8 Viser kvadrerte verdier av måleresultater for "preemption" med 90 % belastning.

	Tale	Video1	Video2	Data1	Data2
1	1,1492E-10	4,4249E-11	1,8171E-08	0,00502965	2074,6203
2	1,1535E-10	4,4223E-11	1,8117E-08	0,00544496	1309,57134
3	1,147E-10	4,4169E-11	1,8171E-08	0,00503248	2531,59923
4	1,1385E-10	4,4156E-11	1,801E-08	0,00511082	1135,42042
5	1,1578E-10	4,3944E-11	1,801E-08	0,00659669	3729,5449
6	1,1513E-10	4,3983E-11	1,801E-08	0,00472200	851,355684
7	1,147E-10	4,4063E-11	1,809E-08	0,00689398	3906
8	1,1535E-10	4,3877E-11	1,809E-08	0,00627739	3989,31192
9	1,13E-10	4,401E-11	1,7796E-08	0,00395138	238,764304
10	1,1449E-10	4,4183E-11	1,8117E-08	0,00696056	3288,56372
<b>M2SUM:</b>	<b>1,1473E-09</b>	<b>4,4086E-10</b>	<b>1,8058E-07</b>	<b>0,05601991</b>	<b>23054,7518</b>



Tabell 5 9 Viser beregning for konfidensgrenser basert på 10 uavhengig observasjoner for "preemption" med 90 % belastning.

	Tale	Video1	Video2	Data1	Data2
$10.0 * m2sum - msum ** 2$ :	1,289E-13	8,341E-15	1,536E-11	0,00426500	24020,8979
$abs(10.0 * m2sum - msum ** 2)/90.0$ :	1,4322E-15	9,2678E-17	1,7067E-13	0,00004700	266,898865
stdev:	3,7845E-08	9,6269E-09	4,1312E-07	0,00685600	16,3370397
konfint:	2,7071E-08	6,8862E-09	2,9551E-07	0,00491700	11,6860022
Nedre grense konfidensgrense:	1,0684E-05	6,6328E-06	0,00013408	0,06964400	33,7591978
Øvre grense konfidensgrensel:	1,0738E-05	6,6466E-06	0,00013468	0,07947800	57,1312022



Figur.5-13 Diagrammet viser sammenligning av ulike kildetyper for "preemptin" med hensyn på forsinkelse. Belastningen er 90 % med ulike seed-verdier.

**Tabell 5-10 Måleresultater etter 10 ganger kjøring av 10 uavhengig observasjoner "for non-preemption" med 90 % belastning.**

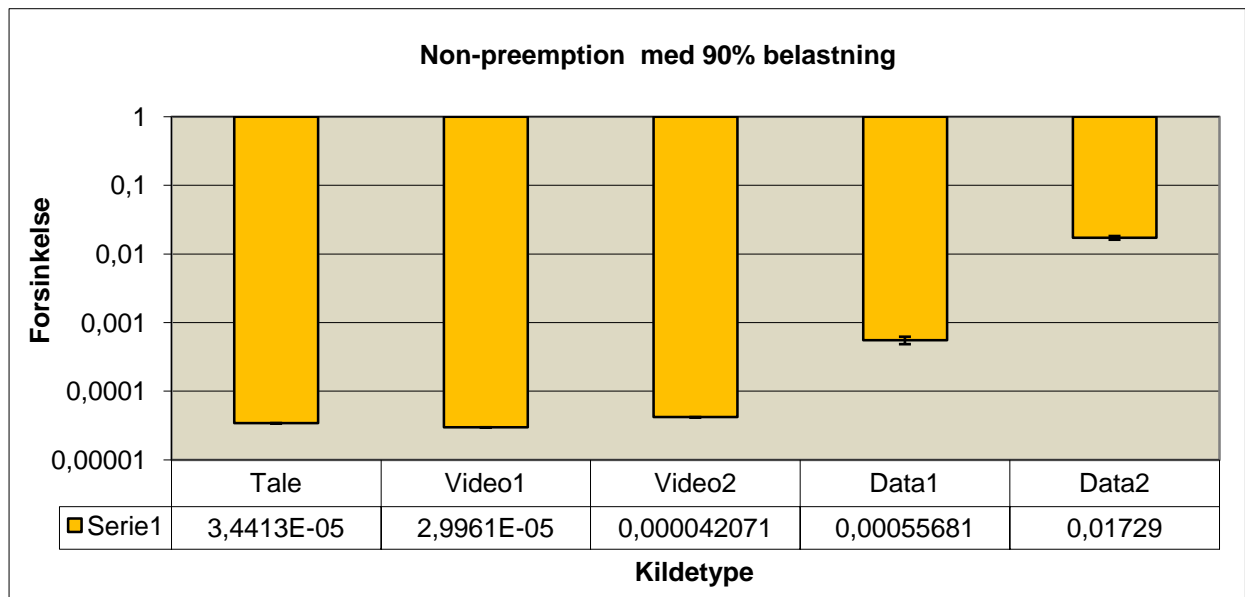
	Tale	Video1	Video2	Data1	Data2
1	3,44E-05	3E-05	4,22E-05	0,000675	0,01682
2	3,45E-05	3E-05	4,23E-05	0,000472	0,01839
3	3,44E-05	3E-05	4,21E-05	0,000538	0,01748
4	3,44E-05	3E-05	4,19E-05	0,000688	0,01906
5	3,45E-05	2,99E-05	4,22E-05	0,000501	0,01675
6	3,43E-05	2,98E-05	4,2E-05	0,000544	0,01564
7	3,45E-05	3,01E-05	4,21E-05	0,000617	0,01858
8	3,45E-05	3E-05	4,22E-05	0,000502	0,01803
9	3,42E-05	2,98E-05	4,16E-05	0,000389	0,0142
10	3,44E-05	3E-05	4,21E-05	0,000641	0,01795
SUM	0,000344	0,0003	0,000421	0,005568	0,1729
<b>Forventning:</b>	<b>3,44E-05</b>	<b>3E-05</b>	<b>4,21E-05</b>	<b>0,000557</b>	<b>0,01729</b>

**Tabell 5-11 Viser kvadrerte verdier av måleresultater for "non-preemption" med 90 % belastning.**

	Tale	Video1	Video2	Data1	Data2
1	8,9643E-11	3,4822E-11	1,1751E-08	2,362E-07	0,00011385
2	9,0364E-11	3,4904E-11	1,1816E-08	2,8998E-07	0,00014689
3	8,9738E-11	3,4621E-11	1,1729E-08	2,809E-07	0,00013502
4	8,9719E-11	3,4751E-11	1,1729E-08	2,2203E-07	0,00012056
5	8,9132E-11	3,4751E-11	1,1707E-08	2,5604E-07	0,00011278
6	9,0117E-11	3,4904E-11	1,1772E-08	3,1125E-07	0,00014762
7	8,9738E-11	3,4786E-11	1,1772E-08	2,0721E-07	0,00011046
8	9,0003E-11	3,4834E-11	1,1751E-08	3,1979E-07	0,00013294
9	8,8925E-11	3,4751E-11	1,1686E-08	3,1159E-07	0,00012455
10	8,8736E-11	3,4786E-11	1,1664E-08	2,6184E-07	9,7496E-05
<b>M2SUM:</b>	<b>8,9612E-10</b>	<b>3,4791E-10</b>	<b>1,1738E-07</b>	<b>2,6968E-06</b>	<b>0,00124218</b>

**Tabell 5-12 Viser beregning for konfidensgrenser basert på 10 uavhengig observasjoner for "non-preemption" med 90 % belastning.**

10.0 * m2sum - msum ** 2:	9,961E-13	6,989E-13	3,9489E-12	8,3936E-07	0,00019809
abs(10.0 * m2sum - msum ** 2)/90.0:	1,1068E-14	7,7656E-15	4,3877E-14	9,3263E-09	2,201E-06
stdev:	1,052E-07	8,8122E-08	2,0947E-07	9,6573E-05	0,00148359
konfint:	7,5253E-08	6,3035E-08	1,4983E-07	6,9079E-05	0,00106122
Nedre grense konfidensgrense:	3,4338E-05	2,9898E-05	4,1921E-05	0,00048773	0,01622878
Øvre grense konfidensgrensel:	3,4488E-05	3,0024E-05	4,2221E-05	0,00062589	0,01835122



**Figur 5-14 Diagrammet viser sammenligning av ulike kildetyper for "non-preemptin" med hensyn på forsinkelse. Belastningen er 90 % med ulike seed-verdier.**

**Tabell 5- 13 Måleresultater etter 10 ganger kjøring av 10 uavhengig observasjoner for "preemption" med 80 % belastning.**

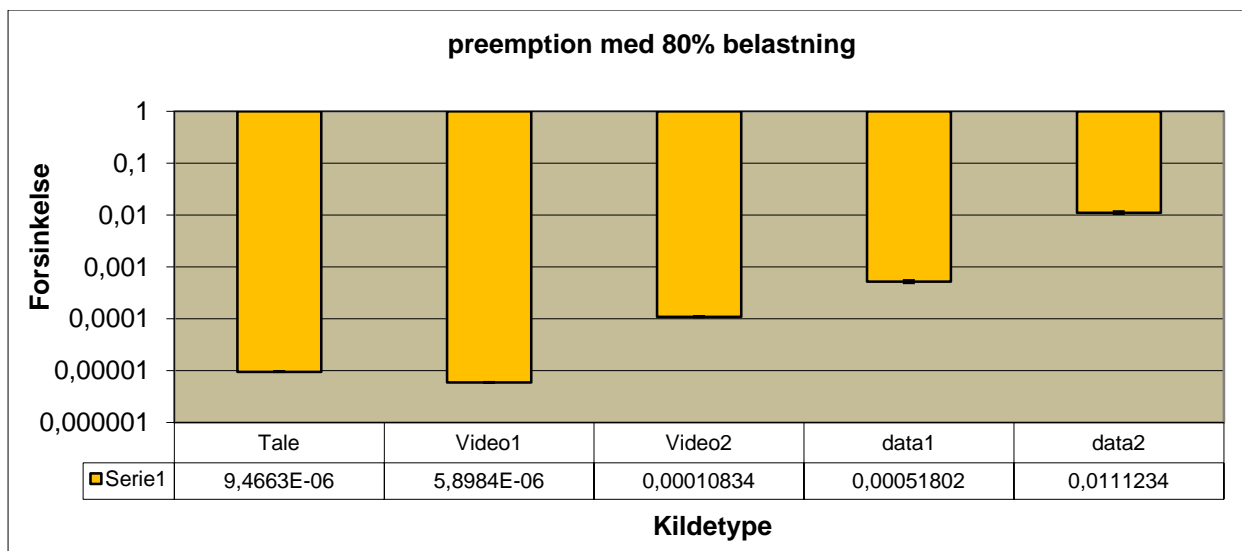
	Tale	Video1	Video2	data1	data2
1	9,47E-06	5,9E-06	0,000108	0,000486	0,01067
2	9,51E-06	5,91E-06	0,000109	0,000539	0,01212
3	9,47E-06	5,88E-06	0,000108	0,00053	0,01162
4	9,47E-06	5,9E-06	0,000108	0,000471	0,01098
5	9,44E-06	5,9E-06	0,000108	0,000506	0,01062
6	9,49E-06	5,91E-06	0,000109	0,000558	0,01215
7	9,47E-06	5,9E-06	0,000109	0,000455	0,01051
8	9,49E-06	5,9E-06	0,000108	0,000566	0,01153
9	9,43E-06	5,9E-06	0,000108	0,000558	0,01116
10	9,42E-06	5,9E-06	0,000108	0,000512	0,009874
SUM	9,47E-05	5,9E-05	0,001083	0,00518	0,111234
<b>Forventning:</b>	<b>9,47E-06</b>	<b>5,9E-06</b>	<b>0,000108</b>	<b>0,000518</b>	<b>0,011123</b>

**Tabell 5 14 Viser kvadrerte verdier av måleresultater for "preemption" med 80 % belastning.**

	Tale	Video1	Video2	Data1	Data2
1	8,9643E-11	3,4822E-11	1,1751E-08	2,362E-07	0,00011385
2	9,0364E-11	3,4904E-11	1,1816E-08	2,8998E-07	0,00014689
3	8,9738E-11	3,4621E-11	1,1729E-08	2,809E-07	0,00013502
4	8,9719E-11	3,4751E-11	1,1729E-08	2,2203E-07	0,00012056
5	8,9132E-11	3,4751E-11	1,1707E-08	2,5604E-07	0,00011278
6	9,0117E-11	3,4904E-11	1,1772E-08	3,1125E-07	0,00014762
7	8,9738E-11	3,4786E-11	1,1772E-08	2,0721E-07	0,00011046
8	9,0003E-11	3,4834E-11	1,1751E-08	3,1979E-07	0,00013294
9	8,8925E-11	3,4751E-11	1,1686E-08	3,1159E-07	0,00012455
10	8,8736E-11	3,4786E-11	1,1664E-08	2,6184E-07	9,7496E-05
<b>M2SUM:</b>	<b>8,9612E-10</b>	<b>3,4791E-10</b>	<b>1,1738E-07</b>	<b>2,6968E-06</b>	<b>0,00124218</b>

**Tabell 5-15** Viser beregning for konfidensgrenser basert på 10 uavhengig observasjoner for "preemption" med 80 % belastning.

	Tale	Video1	Video2	Data1	Data2
<b>10.0 * m2sum - msum ** 2:</b>	6,9441E-14	4,464E-15	3,84E-12	1,337E-07	4,8772E-05
<b>abs(10.0 * m2sum - msum ** 2)/90.0:</b>	7,7157E-16	4,96E-17	4,2667E-14	1,4856E-09	5,4191E-07
<b>stdev:</b>	2,7777E-08	7,0427E-09	2,0656E-07	3,8543E-05	0,00073615
<b>konfint:</b>	1,9869E-08	5,0377E-09	1,4775E-07	2,757E-05	0,00052657
<b>Nedre grense konfidensgrense:</b>	9,4464E-06	5,8934E-06	0,00010819	0,00049045	0,01059683
<b>Øvre grense konfidensgrensel:</b>	9,4862E-06	5,9034E-06	0,00010849	0,00054559	0,01164997



**Figur 5-15** Diagrammet viser sammenligning av ulike kildetyper for "preemtpin" med hensyn på forsinkelse. Belastningen er 80 % med ulike seed-verdier.

**Tabell 5-16 Måleresultater etter 10 ganger kjøring av 10 uavhengig observasjoner for "non-preemption" med 80 % belastning.**

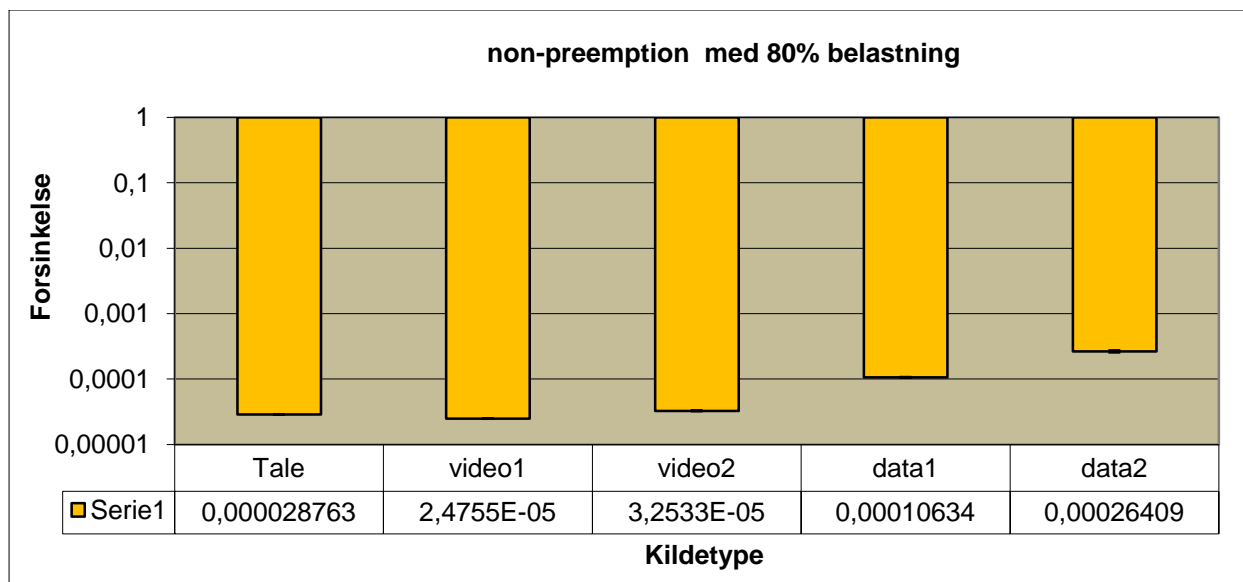
	Tale	video1	video2	data1	data2
1	2,88E-05	2,48E-05	3,22E-05	0,000106	0,000251
2	2,88E-05	2,48E-05	3,26E-05	0,000107	0,000282
3	2,88E-05	2,48E-05	3,26E-05	0,000106	0,000266
4	2,88E-05	2,48E-05	3,26E-05	0,000107	0,000264
5	2,87E-05	2,47E-05	3,26E-05	0,000106	0,000258
6	2,88E-05	2,48E-05	3,25E-05	0,000107	0,000284
7	2,87E-05	2,47E-05	3,25E-05	0,000106	0,000263
8	2,88E-05	2,48E-05	3,26E-05	0,000106	0,000261
9	2,87E-05	2,48E-05	3,25E-05	0,000107	0,000268
10	2,87E-05	2,47E-05	3,25E-05	0,000106	0,000244
SUM	0,000288	0,000248	0,000325	0,001063	0,002641
<b>Forventning:</b>	<b>2,88E-05</b>	<b>2,48E-05</b>	<b>3,25E-05</b>	<b>0,000106</b>	<b>0,000264</b>

**Tabell 5-17 Viser kvadrerte verdier av måleresultater for "non-preemption" med 80 % belastning.**

	Tale	Video1	Video2	Data1	Data2
1	8,2944E-10	6,1504E-10	1,0394E-09	1,1278E-08	6,3152E-08
2	8,3002E-10	6,1306E-10	1,0647E-09	1,1385E-08	7,9637E-08
3	8,2829E-10	6,1405E-10	1,0621E-09	1,13E-08	7,0916E-08
4	8,3059E-10	6,1405E-10	1,0647E-09	1,1406E-08	6,9802E-08
5	8,2541E-10	6,1108E-10	1,0641E-09	1,13E-08	6,6306E-08
6	8,2771E-10	6,1256E-10	1,0589E-09	1,1342E-08	8,0656E-08
7	8,2312E-10	6,1009E-10	1,0556E-09	1,1215E-08	6,9011E-08
8	8,3002E-10	6,1454E-10	1,0628E-09	1,13E-08	6,786E-08
9	8,2541E-10	6,1256E-10	1,0582E-09	1,1342E-08	7,1878E-08
10	8,2312E-10	6,1108E-10	1,0537E-09	1,1215E-08	5,9585E-08
<b>M2SUM:</b>	<b>8,2731E-09</b>	<b>6,1281E-09</b>	<b>1,0584E-08</b>	<b>1,1308E-07</b>	<b>6,988E-07</b>

Tabell 5-18 Viser beregning for konfidensgrenser basert på 10 uavhengig observasjoner for "non-preemption" med 80 % belastning.

	Tale	Video1	Video2	Data1	Data2
$10.0 * m2sum - msum ** 2$ :	2,221E-13	1,005E-13	1,2721E-12	8,04E-12	1,3668E-08
$abs(10.0 * m2sum - msum ** 2)/90.0$ :	2,4678E-15	1,1167E-15	1,4134E-14	8,9333E-14	1,5187E-10
stdev:	4,9677E-08	3,3417E-08	1,1889E-07	2,9889E-07	1,2323E-05
konfint:	3,5534E-08	2,3903E-08	8,5042E-08	2,138E-07	8,815E-06
Nedre grense konfidensgrense:	2,8727E-05	2,4731E-05	3,2448E-05	0,00010613	0,00025528
Øvre grense konfidensgrensel:	2,8799E-05	2,4779E-05	3,2618E-05	0,00010655	0,0002729



Figur 5-16 Diagrammet viser sammenligning av ulike kildetyper for "non-preemptin" med hensyn på forsinkelse. Belastningen er 80 % med ulike seed-verdier.

## 6 Diskusjon og konklusjon

Det er blitt laget en simulering med utgangspunktet i figur 3-11 for å se oppførselen av trafikktyper (tale, video1, video2, data1 og data2) over "DiffServ". Disse trafikktypene er sammenlignet med hensyn på forsinkelse og betjening. Middelerverdier for genererte trafikktyper for 10 ganger kjøring er registrert i tabeller. Det henvises til tabell 5-7, 5-10, 5-13 og 5-16.

### 6.1 Løsningsmetoder for trafikktyper

Det er benyttet to forskjellige løsningsmetoder for å sammenligne trafikktypene med hensyn på forsinkelse og betjening:

- 1a. "preemption" løsning med 90 prosent belastning (se tabell 5-7, 5-8, 5-9 og figur 5-13):  
Registrerte resultater i tabell 5-7 og 5-9 viser at middelerverdi, varians og konfidensintervallet for data1 og spesielt data2 er mye større enn for de andre trafikktypene. Dette varierer mye (fra cirka 15 til cirka 65) på grunn av få observasjoner i løpet av simuleringene. For å få en mer nøyaktig verdi for middelerverdiene kan man kjøre simuleringene lengre.  
Figuren 5-13 viser at data1 og spesielt data2 er blitt veldig forsinket sammenlignet med tale og video1. Dette er forventet fordi det i denne løsningen er brukt både "interrupt" og "Expedited Forwarding (EF)" som har høyere prioritet enn "Assured Forwarding (AF1 og AF2)" og "Best Effort (BE)". Tale bruker enda høyere prioritet, og avbryter data som har lavere prioritet. Med "preemption" kan det forsikres bedre responstid og / eller gjennomstrømming. For mer detaljering om "preemption" henvises til teoridelen under punkt 2.10.
- 2a. "non-preemption" løsning med 90 prosent (se tabell 5-10, 5-11, 5-12 og figur 5-14):  
Her observeres det en del forandringer. Data (spesielt data2) som var veldig forsinket under "preemption", er mye bedre under "non-preemption" (se figur 5-14). Tale, video1 og 2 er nesten like med hensyn på varians, konfidensintervall og forsinkelse (se Tabell 5-10, 5-11, 5-12 og figur 5-14). Det er ikke avbrudd ("interrupt") i dette tilfellet. Det betyr at alle trafikktypene skal betjenes like mye, men tale og video1 har fortsatt høyere prioritet. Dette skyldes at de bruker "EF" som har høyere prioritet enn "AF1", "AF2" og "BE)". Ulempen med "preemption" er at enkelte trafikktyper med lav prioritet kan få for



---

lav "bitrate" og blir aldri betjent. Ved bruk av "non-preemption" blir alle typer behandlet likt, så er man ender ikke opp med trafikktyper som aldri blir transportert.

- 1b. "preemption" løsning med 80 prosent belastning (se tabell 5-13, 5-14, 5-15 og figur 5-15):

Registrerte verdier i tabell 5-15 viser variansen og konfidensintervallet for tale og video1 som er mindre enn for video2, data1 og data2. Figur 5-15 viser også at video2, data1 og data2 er forsinket sammenlignet med tale og video1. Dette er forventet fordi tale bruker "interrupt" i tillegg til at de allerede benytter "EF" som har høyere prioritet enn "AF1", "AF" og "BE". Det er imidlertid viktig å gjøre oppmerksom på at tale er litt dårligere enn video1 med hensyn på forsinkelse.

- 2b. "non-preemption" løsning med 80 prosent belastning (se tabell 5-16, 5-17, 5-18 og figur 5-16):

Ut i fra registrerte verdier i tabell 5-18 er variansen og konfidensintervallet nesten likt for tale, video1 og video2. Det er på grunn av at avbryting "interrupt" ikke er brukt her, og alle trafikktyper betjenes likt, men tale og video1 bruker "EF" som har høyere prioritet enn "AF1", "AF2" og "BE".

Generelt er "preemption" med 80 og 90 prosent belastning bedre for tale og video1 enn "preemption" med hensyn på forsinkelse. Det blir dårligere for video2, data1 og data2.

Verdiene for variansen og konfidensintervallet bekrefter dette.

Spørsmålet er om man bør velge prioritering fremfor betjening eller omvent.

Man kan trekke følgende konklusjon for begge tilfeller:

"preemption": sikrer bedre responstid og / eller gjennomstrømning

"non-preemption": er mindre overhead, mindre "utsulting"<sup>10</sup>

Økning av simuleringstid har ikke hatt noe særlig betydning for forsinkelse, og forfatteren har derfor ikke tatt hensyn til dette. Betjening av trafikktypene er illustrert ved hjelp av histogrammer (se vedlegg 9.2) og utskrift av alle genererte data for kildetyper som er vedlagt på CD'en.

---

<sup>10</sup> "utsulting" (starvation): Hvis det kommer en data fortløpende med høyere prioritet og avbryter data med lavere prioritet, kan det hende at data med lavere prioritet aldri blir betjent [79].

"Jitter" er nevnt i oppgaven som en av policyer, men ble ikke undersøkt videre på grunn av tidsbegrensning.

Det er en mulighet i programmet for å sette tredje policy/ strategi, slik at alle trafikktyper kan avbryte hverandre som for eksempel tale avbryter video1, video1 avbryter video2, video2 avbryter data1 og data1 avbryter data2. Dette kan gjøres ved å fjerne "pri = 1" i programmet og skrive "if presim and" istedenfor. Men dette er ikke gjort på grunn av tidsbegrensning.

## 6.2 Statistisk registrering

Vi har lagt inn et delprogram i programmet for å registrere totalt antall pakker ("amount \_packets") og antall pakker for (tale, video1, video2, data1 og data2) som genereres. Dette ble gjort for å regne ut hvor stor del av kapasiteten er blitt brukt under hver kjøring. En kan beregne totalt kapasiteten ved å dele totalt antall pakker ("amount \_packets") på simuleringstid.

## 6.3 Feilkilder

Generering av trafikkilder (tale, video1, video2, data1 og data2) over "DiffServ" ble gjort og simulert lokalt på en datamaskin.

Trafikkilder i denne oppgaven er ikke store og kompliserte, derfor kan feil korrigeres fort.

Programmet er kjørt på en bærbar PC med 2,5 GB RAM, og det har oppstått problemer under kjøring av programmet med minne oppbrukt. For å løse dette problemet kan man skrive uttrykket: "cim -m1500" og navnet til programmet. I denne oppgaven er navnet til programmet "sim-v3-aso-a-09".

## 2.6 Programmeringsspråk

I denne oppgaven ble "*simulation language* (SIMULA<sup>11</sup>)" valgt og brukt som simuleringsverktøy. Se vedlegg 9.1 for programkode.

---

<sup>11</sup> Simula var det første objektorienterte programmeringsspråket, og en forgjenger for språkene "Smalltalk" og C++.

Simula ble utviklet for simuleringer, og behovet innenfor det feltet dannet rammen for mange av aspektene ved objektorienterte programmeringsspråk i dag, inkludert virtuelle funksjoner. Simula ble utviklet i 1967 ved Norsk Regnesentral med Ole-Johan Dahl og Kristen Nygaard i spissen [58].

Det ble tatt utgangspunkt i kode for en talekilde fra "STOLAS" [69]. Den ble endret/tilpasset og brukt i denne oppgaven.

Utskrift av data for middelveier til kildetyperne (tale, video1, video2, data1 og data2) etter 10 ganger kjøring for "preemption" og "non-preemption" med 80 og 90 prosent belastning er vedlagt på en CD. I tillegg er det vedlagt utskrift for to kjøring (se vedlegg 9.2).

Se også vedlegg 9.4 for oppsett og kjøring av programmet.

## 7 Videre arbeid

Mye arbeid og kartlegging gjenstår før en får svar på hvordan "DiffServ" kan garantere tjenestekvalitet. Jeg tenker da først og fremst på hvordan sikkerheten ivaretas i forbindelse med tap, forsinkelse og "jitter" i "DiffServ". Det kan være en god idé å gjøre en større simulering for å se hvordan trafikkildene (tale, video, og data) fungerer over "DiffServ" i praksis.

Anbefales også generering av flere forskjellig kilder og bruke protokollene "MPLS og GMPLS" over "DiffServ".

---

## 8 Litteraturreferanser

- [1] ZHU, H., LI, M., CHLMTAC, I., PRABHAKARAN, B., (2004). A SURVEY OF QUALITY OF SERVICE IN IEEE 802.11 NETWORKS. IEEE Wireless Communications, 1536-1284/04, Side: 6-14.
- [2] Shin, J., Kim, J. W., Jay Kuo, C.-C., (2001). Quality-of-Service Mapping Mechanism for Packet Video in Differentiated Services Network. IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 3, NO. 2, Side: 219-231.
- [3] <http://www.objs.com/survey/QoS.htm>, (dato: 03.04.2009).
- [4] [http://en.wikipedia.org/wiki/Quality\\_of\\_service](http://en.wikipedia.org/wiki/Quality_of_service), (dato: 22.03.2009).
- [5] Dimitrios P. P., Hutchison, D., Quality of Service Assurance for the Next Generation Internet. Computing Department Faculty of Applied Sciences Lancaster University Lancaster, UK LA1 4YR.
- [6] Peder, J. E., Poul, E. H., Bjarne, E. H. (2000). Pålitelighet og ytelse i informasjons- og kommunikasjonssystem Grunnlag. Institut for telematikk [NTNU], Side: 1-266.
- [7] Janusz, G., Andrzej J., Rafal, S. (2003). Quality of Service Terminology in IP Networks. IEEE Communications Magazine. 0163-6804/03, Side: 153-159.
- [8] Norsk riks kingkasting, mars 2009. <http://www1.nrk.no/nett-tv>, (dato: 06.04.2009).
- [9] Tandberg website, mars 2009. <http://www.tanberg.net>, (dato: 09.04.2009).
- [10] Aso Amedyan, Protokoller for pålitelig dataoverføring pakkebaserte optiske nett, Norges Teknisk-Naturvitenskaplige Universitet, 2008.

- 
- [11] Network Working Group J. polk Request for Comments: 4411 Cisco Systemes, Kategori: Standards Track februar 2006.
- [12] S. Blake Torrent Networking Technologies D. Black EMC Corporation M. Carlson Sun Microsystems E. Davies Nortel UK Z. Wang Bell Labs Lucent Technologies W. Weiss Lucent Technologies December 1998 “Network Working Group Request for Comments: 2475 Category: Informational.
- [13] Youngtak, K., Eunhyuk, L., Douglas, M. (2002). Discrete Event Simulation of the DiffServ-over-MPLS with NIST GMPLS Lightwave Agile. Dept. of Information and Communication Engineering, Graduate School, YeungNam Univ. JCCI2002.
- [14] Terrasa S., S´aez, S., Vila, J., Hern´andez, E. Comparing the utilization bounds of IntServ and DiffServ. Universidad Polit´ecnica de Valencia, C/Camino de Vera s/n 46022 Valencia, SPAIN. Side: 1-10
- [15] Un Yoon, S., Hoon Lee, J., Sun Lee, K., Hee Kang, C. (2000). QoS Support in MobileWireless IP Networks using Differentiated Services and Fast Handoff Method. Broadband Telecommunication Laboratory Department of Electronics Engineering, Korea University. Side: 266-270.
- [16] Youngtak, K., Chul K. (2003). QoS-Guaranteed DiffServ-Aware-MPLS Traffic Engineering with Controlled Bandwidth Borrowing. C.V. Ramamoorthy, R.Y. Lee, and K.W. Lee (Eds.): SERA 2003, LNCS 3026, pp. 253–265, 2004.
- [17] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W. (December 1998). RFC2475 - An Architecture for Differentiated Service. Network Working Group, Request for Comments: 2475, Category: Informational.
- [18] <http://www.faqs.org/rfcs/rfc2597.html>, (dato: 09.04.2009).
- [19] <http://qbone.internet2.edu/bb/Bucket.doc>, (dato: 25.05.2009).

- 
- [20] <http://www.faqs.org/rfcs/rfc2638.html>, (dato: 23.05.2009).
- [21] Jacobson, V., Nichols, K., Poduri, K. An expedited forwarding phb, RFC 2598, 1999.
- [27] Heinanen, J., Baker, F., Weiss, W., Wroclawski, J. Assured forward phb group, RFC 2597, 1999.
- [28] Serban, R., Barakat, C., Dabbous, W. (2004). A CBQ-Based Dynamic Resource Allocation Mechanism for Diffserv Routers. Planète Project, INRIA Sophia Antipolis.
- [29] MOHD NAZRI BIN MOHD WARIP. (2007). QUALITY OF SERVICE AND TRAFFIC PREEMPTION FOR MULTI-PROTOCOL LABEL SWITCHING / GENERALIZED MULTIPROTOCOL LABEL SWITCHING NETWORKS. Faculty of Electrical Engineering Universiti Teknologi Malaysia.
- [30] [http://en.wikipedia.org/wiki/Preemption\\_\(computing\)](http://en.wikipedia.org/wiki/Preemption_(computing)), (Dato:16.03. 2009).
- [31] Cerf, V., (september 1981). RFC794 – Preemption. Network Working Group, Request for Comments: 794, Replaces: IEN 125.
- [32] <http://en.wikipedia.org/wiki/AUTOVON>, (dato: 17.03. 2009).
- [33] [http://www.cisco.com/en/US/tech/tk543/tk757/technologies\\_tech\\_note09186a00800949f2.html](http://www.cisco.com/en/US/tech/tk543/tk757/technologies_tech_note09186a00800949f2.html), (Dato: 20.03.2009).
- [34] Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA. (2007). Overview of DiffServ for Quality of Service. Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2.
- [35] <http://www.streaming-list.net/interactive.php3>, (Dato: 25.03.2009).

- 
- [36] Burns, A., Wellings, A., Third Edition (2001). Real-Time Systems and Programming Languages. Side: 1-738.
- [36] [http://searchnetworking.techtarget.com/sDefinition/0,,sid7\\_gci1069410,00.html](http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci1069410,00.html), (Dato: 27.03.2009).
- [37] <http://forskningsnett.uninett.no/uninytt/1999-1/qos.html>, (Dato: 22.03.2009).
- [38] <http://www.ipkompetanse.no/moderne/oh/IP-nett-utg2-kap3-delA.ppt#353,12,Prioritetskøer>, (Dato: 22.03.2009).
- [39] Yang, M., Wang, J., Lu, E., Zheng, S. Q. (2004). Hierarchical Scheduling for DiffServ. Department of Electrical and Computer Engineering, University of Nevada Las Vegas, Las Vegas, NV 89154, Department of Computer Sciences, Georgia Southern University, Statesboro, GA 30460, Dept. of Mathematics and Computer Science, Salisbury University, Salisbury, MD 21801, Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75080 Classes.
- [40] <http://forskningsnett.uninett.no/uninytt/1999-1/qos.html>, (Dato: 20.03.2009).
- [41] <http://domen.uninett.no/~oak/foredrag/qos/qos-unin.txt>, (Dato: 21.03.2009).  
, (Dato: 20.03.2009).
- [42] [http://www.cisco.com/en/US/docs/ios/12\\_2/qos/configuration/guide/qcfdfsrv\\_ps1835\\_TSD\\_Products\\_Configuration\\_Guide\\_Chapter.html](http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfdfsrv_ps1835_TSD_Products_Configuration_Guide_Chapter.html), (Dato: 20.03.2009).
- [43] [http://en.wikipedia.org/wiki/Transport\\_network](http://en.wikipedia.org/wiki/Transport_network), (Dato: 15.03.2009).
- [44] Ling, X., Kuang-Hao Liu, Cheng, Y., Xuemin (Sherman) Shen and Jon W. Mark. (2007). A Novel Performance Model for Distributed Prioritized MAC Protocols. IEEE Xplore. Downloaded on March 27, 2009 at 08:33 from IEEE Xplore. Restrictions apply. Side: 4692-4696.



- [45] <http://www.cisco.com/en/US/docs/cable/cmts/feature/ubrTOSow.html#wp1057508>, (Dato: 28.03.2009).
- [46] <http://www.networkworld.com/details/661.html?def>, (Dato: 28.03.2009).
- [47] [http://ftp.utcluj.ro/pub/users/tarc/Doc/mps\\_qos.ppt#321,14,IPv4 ToS vs. DS-Field](http://ftp.utcluj.ro/pub/users/tarc/Doc/mps_qos.ppt#321,14,IPv4>ToS>vs.>DS-Field), (Dato: 29.03.2009).
- [48] <http://networknerd.wordpress.com/2009/01/20/dscp-to-type-of-service-mappings/>, (Dato: 29.03.2009).
- [49] <http://www.lancope.com/solutions/networkoperations/QualityOfService.aspx>, (Dato: 29.03.2009).
- [50] Peder J. Emstad, Poul E. Heegaard, Bjarne E. Helvik. (2000). Pålitelighet og ytelse i informasjons- og kommunikasjonssystem Grunnlag. Institutt for telematikk (NTNU). (Side: 1-266).
- [51] [http://www.cisco.com/en/US/docs/ios/qos/configuration/guide/dffsrv\\_for\\_qos\\_oview.pdf](http://www.cisco.com/en/US/docs/ios/qos/configuration/guide/dffsrv_for_qos_oview.pdf), (Dato: 09.03.2009).
- [52] [http://whatis.techtarget.com/definition/0,,sid9\\_gci523729,00.html](http://whatis.techtarget.com/definition/0,,sid9_gci523729,00.html) (dato: 24.03.2009).
- [53] [http://en.wikipedia.org/wiki/Type\\_of\\_Service](http://en.wikipedia.org/wiki/Type_of_Service) (dato: 25.03.2009).
- [54] <http://www.rfc-editor.org/rfc/rfc791.txt> (dato: 22.03.2009).
- [55] <http://www.nlm.nih.gov/medlineplus/surgeryvideos.html> (dato: 28.02.2009).
- [56] <http://domen.uninett.no/~oak/foredrag/qos/qos-unin.txt>, (Dato: 29.03.2009)
- [57] [http://en.wikipedia.org/wiki/Ns\\_\(simulator\)](http://en.wikipedia.org/wiki/Ns_(simulator)), (dato: 28.04.2009).

- [58] <http://no.wikipedia.org/wiki/Simula> (dato: 12.03.2009).
- [59] <http://www.freesoft.org/CIE/RFC/1812/111.htm> (dato: 24.04.2009).
- [60] <http://windowshelp.microsoft.com/Windows/nb-NO/help/7ced6d16-fb98-475f-aaca-87f820b3a9871044.msp>, (dato: 08.05.2009).
- [61] <http://orakelet.info/5560>, (dato: 08.05.2009).
- [62] [http://searchunifiedcommunications.techtarget.com/tip/0,289483,sid186\\_gci1032455,00.htm](http://searchunifiedcommunications.techtarget.com/tip/0,289483,sid186_gci1032455,00.htm), (Dato: 22.04.2009).
- [63] [http://www.cisco.com/en/US/docs/ios/12\\_0/qos/configuration/guide/qcpolts.html](http://www.cisco.com/en/US/docs/ios/12_0/qos/configuration/guide/qcpolts.html), (Dato: 22.04.2009).
- [64] [http://www.cisco.com/en/US/tech/tk543/tk545/technologies\\_tech\\_note09186a00800a3a25.shtml](http://www.cisco.com/en/US/tech/tk543/tk545/technologies_tech_note09186a00800a3a25.shtml), (Dato: 25.04.2009)
- [65] <http://www.faqs.org/rfcs/rfc2212.html>, (dato: 22.05.2009).
- [66] Birtwistle, G., (2003). Demos - a system for Discrete Event Modelling on Simula. School of Computer Science, University of Sheffield, Regent Court, 211 Portobello Street, Sheffield, England S1 4DP. Side: 1-238.
- [67] INTERNATIONAL TELECOMMUNICATION UNION, TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU. (03.93). TELEPHONE TRANSMISSION QUALITY OBJECTIVE MEASURING APPARATUS. ARTIFICIAL CONVERSATIONAL SPEECH. ITU-T Recommendation P.59. (Previously "CCITT Recommendation").
- [68] Jahangir, D., C. (April, 2008). Internet Traffic Modeling and Capacity Evaluation in UMTS. International Journal of Hybrid Information Technology. Side: 109-120.

- 
- [69] Ragnar, Ø. A., Jan, C., Rolf, B. H., Bjarne E. H., Ton, K., Christophe, P., Astrid, S., Norvald, S., Aasmund, S., Idelfonso, T. M., Erik, V. B., Evi, Z. (2001). Switching Technologies for Optically Labeled Signals. IST-2000-28557 STOLAS. Side: 1-118.
- [70] Georgoulas, S., Pavlou, G., Flegkas, P., Trimintzios, P. Buffer and Bandwidth Management for the Expedited Forwarding Traffic Class in Differentiated Services Networks. Centre for Communication Systems Research, University of Surrey, UK.
- [71] Chimeh, J. D., Internet Traffic Modeling and Capacity Evaluation in UMTS. (2008). International Journal of Hybrid Information Technology. Vol. 1, No. 2, Side: 109-120.
- [72] Peder J. Emstad, Poul E. Heegaard, Bjarne E. Helvik. Pålitelighet og ytelse i informasjons- og kommunikasjonssystem Grunnlag. (2000-08-11). Institutt for telematikk. Side: 1-266.
- [73] <http://www.iet.ntnu.no/projects/beats/Documents/Rapport.pdf>, (dato: 18.06.2009).
- [74] <https://www.sykepleierforbundet.no/ikbViewer/Content/112911/Trond%20Harald%20Olsen%20%20Simulering%20i%20helsevesenet.pdf>, (Dato: 26.4.2009).
- [75] <http://no.wikipedia.org/wiki/Gigabyte>, (dato: 38.06.2009).
- [76] Niblett, B., Rohl, J.S., Whitfield, H. Simulation techniques for discrete event systems. (1988). CAMBRIDGE UNIVERSITY PRESS. Side: 1-185.
- [77] [http://en.wikipedia.org/wiki/Student's\\_t-distribution](http://en.wikipedia.org/wiki/Student's_t-distribution), (dato: 17.07.2009).
- [78] <http://cnx.org/content/m13406/latest/>, (dato: 04.23.2009).

[79] [http://en.wikipedia.org/wiki/Starvation\\_\(computers\)](http://en.wikipedia.org/wiki/Starvation_(computers)), (dato: 26.6.2009).



---

```

    nevner := (numb_sources-numb_active)*lambda + numb_active*my;
    time_to_next := -(LN(trukket))/nevner;
comment ** generering av alle pakker i perioden til neste tilstansshift
8.33333 er en pakkerate av en pakke som er gen. pr. kilde **;
    temp := numb_active * 8.33333 * time_to_next;
    numb_long := Entier(temp);
    frac_long := temp - Entier(temp);
comment ** Finner intervall (mellomtid el. pause) mellom lange pakker pluss fraksjoner**;
time_spacing := time_to_next/(numb_long + 1);
for i:=1 step 1 until numb_long do
begin
if pakkelager.length > 0 then
begin
    pck:-pakkelager.coopt;
    pck.q:-q;
    pck.size:=7680;
    pck.s_type:=1;
    pck.schedule((i-1)*time_spacing);
end
else NEW Packet("pack", q, 7680, s_type).schedule((i-1)*time_spacing);
% *****
% comment ** STATISTICS: count IP packets gen. globally **;
    tot_packets := tot_packets + 1;
    amount_packets := amount_packets + 7680;
    amount_voice := amount_voice + 7680;
% *****
end ** long packets **;
comment ** sender ut sist el. delvis av langepakker **;
if frac_long > 0.000000001 then
begin
if pakkelager.length > 0 then
begin
    pck:-pakkelager.coopt;
    pck.q:-q;
    pck.size:=frac_long*7680;
    pck.s_type:=1;
    pck.schedule(numb_long*time_spacing);
end
else NEW Packet("pack", q, frac_long*7680, s_type).schedule(numb_long*time_spacing);
end;
% *****
% comment ** STATISTICS: count IP packets gen. globally **;
    tot_packets := tot_packets + 1;
    amount_packets := amount_packets + frac_long*7680;

```



---

```

begin
  pck:-pakkelager.coopt;
  pck.q:-q;
  pck.size:=12000;
  pck.s_type:= s_type;
  pck.schedule((i-1)*time_spacing);
end
else NEW Packet("pack",q,12000, s_type).schedule((i-1)*time_spacing);
% *****%
comment ** STATISTICS: count IP packets gen. globally **;
  tot_packets := tot_packets + 1;
  amount_packets := amount_packets + 12000;
  amount_video1 := amount_video1 + 12000;
% *****
end ** long packets **;
comment ** sender ut sist el. delvis av langepakker **;
if frac_long > 0.000000001 then
begin
if pakkelager.length > 0 then
begin
  pck:-pakkelager.coopt;
  pck.q:-q;
  pck.size:=frac_long*12000;
  pck.s_type:= s_type;
  pck.schedule(numb_long*time_spacing);
end
  else NEW Packet("pack", q, frac_long*12000, s_type).schedule(numb_long*time_spacing);
end;
% *****%
comment ** STATISTICS: count IP packets gen. globally **;
  tot_packets := tot_packets + 1;
  amount_packets := amount_packets + frac_long*12000;
  amount_video1 := amount_video1 + frac_long*12000;

% *****
comment ** beholder (fortsetter) til neste tilstandsskift i "birth-death" prosess **;
  hold(time_to_next);
comment ** draw if increase or decrease in number of active sources **;
if Flatford.sample < ((numb_sources-numb_active)*lambda)/nevner then
numb_active := numb_active + 1
else numb_active := numb_active - 1;
  END ** while **;
END ** class video1 (interaktiv) **;
% *****

```





---

```

comment ** STATISTICS: count IP packets gen. globally **;
tot_packets := tot_packets + 1;
amount_packets := amount_packets + 12000;
amount_video2 := amount_video2 + 12000;
% *****
end ** long packets **;
comment ** sender ut sist el. delvis av langepakker **;
if frac_long > 0.000000001 then
begin
if pakkelager.length > 0 then
begin
pck:-pakkelager.coopt;
pck.q:-q;
pck.size:=frac_long*12000;
pck.s_type:= s_type;
pck.schedule(numb_long*time_spacing);
end
else NEW Packet("pack", q, frac_long*12000, s_type).schedule(numb_long*time_spacing);
end;
% *****
% comment ** STATISTICS: count IP packets gen. globally **;
tot_packets := tot_packets + 1;
amount_packets := amount_packets + frac_long*12000;
amount_video2 := amount_video2 + frac_long*12000;
% *****
comment ** beholder (fortsetter) til neste tilstandsskift i "birth-death" prosess **;
hold(time_to_next);
comment ** draw if increase or decrease in number of active sources **;
if Flatford.sample < ((numb_sources-numb_active)*lambda)/nevner then
numb_active := numb_active + 1
else numb_active := numb_active - 1;
END ** while **;
END ** class video2 (streaming) **;
% *****

% *****
comment ** Simulering for daata1 (ftp) **;
COMMENT ** Generering av ftp-trafikkilde **;
entity class data1(q);
REF(waitq) q;
BEGIN
ref(packet) pck;
integer i, numb_active, numb_long, s_type, size;
long real trukket, nevner, time_to_next, frac_long, temp, time_spacing, lambda, my;

```



---

```

    pck.q:-q;
    pck.size:=frac_long*3840;
    pck.s_type:= s_type;
    pck.schedule(numb_long*time_spacing);
end
else NEW Packet("pack", q, frac_long*3840, s_type).schedule(numb_long*time_spacing);
end;
% *****
% comment ** STATISTICS: count IP packets gen. globally **;
    tot_packets := tot_packets + 1;
    amount_packets := amount_packets + frac_long*3840;
    amount_data1 := amount_data1 + frac_long*3840;
% *****
comment ** beholder (fortsetter) til neste tilstandsskift i "birth-death" prosess **;
hold(time_to_next);
comment ** draw if increase or decrease in number of active sources **;
if Flatford.sample < ((1-numb_active)*lambda)/nevner then
numb_active := numb_active + 1
else numb_active := numb_active - 1;
    END ** while **;
END ** class data1 (ftp) **;
% *****

% *****
comment ** Simulering for data2 (e-mail) **;
COMMENT ** Generering av en e-meil-trafikkilde **;
entity class data2(q);
% integer numb_sources;
REF(waitq) q;
BEGIN
    ref(packet) pck;
    integer i, numb_active, numb_long, s_type, size;
    long real trukket, nevner, time_to_next, frac_long,
    temp, time_spacing, lambda, my;
comment ** s_type kan brukes i burster for stat.reg. formåler**;
s_type := 5;
lambda:= 1/90.8;
my:= 1/0.3;
if Flatford.sample < 0.003293 then numb_active := 1
else numb_active := 0;
WHILE TRUE DO
BEGIN
    trukket := Flatford.sample;
    if trukket=0.0 then trukket:= 0.000000000000000000000001;

```

---

```

    nevner := (1-numb_active)*lambda + numb_active*my;
    time_to_next := -(LN(trukket))/nevner;
comment ** generering av alle pakker i perioden til neste tilstansshift
100 er en rate av pakken gen. pr. kilde **;
    temp := numb_active * 100 * time_to_next;
    numb_long := Entier(temp);
    frac_long := temp - Entier(temp);
comment ** Finner intervall (mellomtid el. pause) mellom lange pakker pluss fraksjon**;
time_spacing := time_to_next/(numb_long + 1);
for i:=1 step 1 until numb_long do
begin
if pakkeler.length > 0 then
begin
    pck:-pakkeler.coopt;
    pck.q:-q;
    pck.size:=3840;
    pck.s_type:= s_type;
    pck.schedule((i-1)*time_spacing);
end
else NEW Packet("pack", q, 3840, s_type).schedule((i-1)*time_spacing);
% *****
% comment ** STATISTICS: count IP packets gen. globally **;
    tot_packets := tot_packets + 1;
    amount_packets := amount_packets + 3840;
    amount_data2 := amount_data2 + 3840;
% *****
end ** long packets **;
comment ** sender ut sist el. delvis av langepakker **;
if frac_long > 0.00000001 then
begin
if pakkeler.length > 0 then
begin
    pck:-pakkeler.coopt;
    pck.q:-q;
    pck.size:=frac_long*3840;
    pck.s_type:= s_type;
    pck.schedule(numb_long*time_spacing);
end
else NEW Packet("pack", q, frac_long*3840, s_type).schedule(numb_long*time_spacing);
end;
% *****
% comment ** STATISTICS: count IP packets gen. globally **;
    tot_packets := tot_packets + 1;
    amount_packets := amount_packets + frac_long*3840;

```

---

```

    amount_data2 := amount_data2 + frac_long*3840;
% *****
comment ** beholder (fortsetter) til neste tilstandsskift i "birth-death" prosess **;
hold(time_to_next);
comment ** draw if increase or decrease in number of active sources **;
if Flatford.sample < ((1-numb_active)*lambda)/nevner then
numb_active := numb_active + 1
else numb_active := numb_active - 1;
    END ** while **;
END ** class data2 (e-meil) **;
% *****

% *****
entity class packet (q, size, s_type);
    ref (waitq) q;
    integer size, s_type;
begin
    integer pri;
    long real start_paaka;
start:
    start_paaka:=time;
    if s_type=1 or s_type=2 then pri := 1
    else if s_type=3 then pri := 2
    else if s_type=4 then pri := 3
    else pri := 4;
% Alternativt hvis ønsker at alle (ikke kun pri=1) skal kunne avbryte alle lavere prioriteter:
% if pri < betjener.pri and not betjener.ledig then
    if presim and pri=1 and pri < betjener.pri and not betjener.ledig then
        begin
            betjener.interrupt(1);
% For debug:
%     outtext("Tid:");
%     outfix(time,4,8);
%     outtext(" Pakke-interrupt med pri= ");
%     outint(pri,2);
%     outtext(" / Betjener.pri= ");
%     outint(betjener.pri, 2);
%     outimage;
        end;
        q_sync.give(1);
        q.wait;
        pakkelager.wait;
        goto start;
end;
end;

```

```
entity class diff_serv;
begin
  boolean ledig;
  integer pri;
  long real stid_betj;
  ref(packet) pek;
  start:
    ledig := true;
    q_sync.take(1);
    ledig := false;
    stid_betj := time;
    if q(1).length > 0 then
      begin
        pri := 1;
        pek := q(1).first;
        hold(pek.size/line_speed);
        if interrupted = 0 then pek:-q(1).coopt
        else
          begin
            wasted_s_time(1) := wasted_s_time(1) + (time - stid_betj);
            interrupted := 0;
            q_sync.give(1);
            pek := none;
          end;
        end
      else if q(2).length > 0 then
        begin
          pri := 2;
          pek := q(2).first;
          hold(pek.size/line_speed);
          if interrupted = 0 then pek:-q(2).coopt
          else
            begin
              wasted_s_time(2) := wasted_s_time(2) + (time - stid_betj);
              interrupted := 0;
              q_sync.give(1);
              pek := none;
            end;
          end
        else if q(3).length > 0 then
          begin
            pri := 3;
            pek := q(3).first;
            hold(pek.size/line_speed);
```

---

```
    if interrupted = 0 then pek:-q(3).coopt
    else
    begin
        wasted_s_time(3) := wasted_s_time(3) + (time - stid_betj);
        interrupted := 0;
        q_sync.give(1);
        pek :- none;
    end;
end
else if q(4).length > 0 then
begin
    pri := 4;
    pek :- q(4).first;
    hold(pek.size/line_speed);
    if interrupted = 0 then pek:-q(4).coopt
    else
    begin
        wasted_s_time(4) := wasted_s_time(4) + (time - stid_betj);
        interrupted := 0;
        q_sync.give(1);
        pek :- none;
    end;
end;
if pek /= none then
begin
%   p_delay(pek.s_type).update(time-pek.start_paaka);
    p_delay(pek.s_type).update(time-pek.size/line_speed-pek.start_paaka);
    pek.schedule(0.0);
    pek:-none;
end;
goto start;
end;

COMMENT ** Hovedprogram **;
seedverdi:=907;
setseed(seedverdi);
Flatford  :- new uniform("Flatford", 0.0, 1.0);
pakkelager :-new waitQ ("pakker");
line_speed := 100000000.0;
tot_tale  :=16;
tot_video1 :=16;
tot_video2 :=16;
tot_data1 :=28.8;
tot_data2 :=23.2;
```



---

```

q(1)  :-new waitq ("EF");
q(2)  :-new waitq ("AF1");
q(3)  :-new waitq ("AF2");
q(4)  :-new waitq ("BE");
q_sync :-new bin ("SYNC", 0);
p_delay(1) :-new histogram("tale", 0.0, 0.001, 16);
p_delay(2) :-new histogram("video1", 0.0, 0.001, 16);
p_delay(3) :-new histogram("video2", 0.0, 0.001, 16);
p_delay(4) :-new histogram("data1", 0.0, 0.001, 28.8);
p_delay(5) :-new histogram("data2", 0.0, 0.001, 3.2);
betjener :- new diff_serv("diff_serv");
betjener.schedule(0.0);
new speech("tale", tot_tale, q(1)).schedule(0.0);
new video1("interaktiv", tot_video1, q(1)).schedule(0.0);
new video2("streaming", tot_video2, q(2)).schedule(0.0);
for k:=1 step 1 until tot_data1 do
new data1("ftp", q(3)).schedule(0.0);
for k:=1 step 1 until tot_data2 do
new data2("e_mail", q(4)).schedule(0.0);
trace;
presim := false;
tracetid := 0.05;
simtid := 10000.0;
hold(tracetid);
notrace;
COMMENT ** kjøre i 1000 sekunder eller 10000 **;
hold (simtid);
%*****
COMMENT** beregning av
msum:=0.00000000000009961;
m2sum:=0.000000315611;
stdev := sqrt(abs(10.0 * m2sum - msum ** 2)/90.0);
konfint := stdev * 1.833 / sqrt(10.0);
% *****
if presim then outtext("Preemptiv simulering!")
else outtext("Non-preemptiv simulering!");
outimage; outimage;
outtext("konfint=");
outfix(konfint, 2, 22);
outimage;
outimage;

outtext("simtid=");
outfix(simtid, 2, 22);
outimage;

```

```
outimage;  
outimage;  
outtext("seedverdi=");  
outint(seedverdi, 22);  
outimage;  
outimage;
```

```
outimage;  
outtext("tot_packets=");  
outint(tot_packets, 22);  
outimage;  
outimage;
```

```
outtext("amount_packets=");  
outfix(amount_packets, 2, 22);  
outimage; outimage;
```

```
outtext("amount_voice=");  
outfix(amount_voice, 2, 22);  
outimage;  
outimage;
```

```
outtext("amount_video1=");  
outfix(amount_video1, 2, 22);  
outimage;  
outimage;
```

```
outtext("amount_video2=");  
outfix(amount_video2, 2, 22);  
outimage;  
outimage;
```

```
outtext("amount_data1=");  
outfix(amount_data1, 2, 22);  
outimage;  
outimage;
```

```
outtext("amount_data2=");  
outfix(amount_data2, 2, 22);  
outimage;  
outimage;
```

```
outtext("Wasted service time:");  
outimage;
```

---

```

outtext("EF : "); outfix(wasted_s_time(1), 8, 22);
outimage;
outtext("AF1 : "); outfix(wasted_s_time(2), 8, 22);
outimage;
outtext("AF2 : "); outfix(wasted_s_time(3), 8, 22);
outimage;
outtext("BE : "); outfix(wasted_s_time(4), 8, 22);
outimage;
outimage;
% *****
COMMENT ** brukes transient for å la simulatoren oppnå stabil tilstand **;
% hold(transtime);
COMMENT ** Alle statistikker registreres hvis tid > transtime **;
COMMENT ** Dette er for DEMOS statistikk: **;
% reset;
% COMMENT ** Starter simulering **;
% COMMENT ** feilsparing av start **;
    end;
end;

```

## 9.2 Utskrift av genererte data for kildetyper

### Utskrift av "preemption med 80 prosent belastning:

```

                                CLOCK TIME = 5.000&-002
*****
*                                                                           *
*           T R A C I N G   S W I T C H E D   O F F                       *
*                                                                           *
*****

Preemptiv simulering!

simtid=                5000.00

seedverdi=              10000

tot_packets=           68345017

amount_packets=        399396628172.98

amount_voice=          79914701190.42

amount_video1=         79534206152.47

```

---

amount\_video2= 79834795689.28  
amount\_data1= 144156946300.16  
amount\_data2= 15955978840.64

Wasted service time:  
EF : 0.00000000  
AF1: 194.54172809  
AF2: 133.78343355  
BE : 19.26107659

CLOCK TIME = 5000.050

\*\*\*\*\*  
\* R E P O R T \*  
\*  
\*\*\*\*\*

D I S T R I B U T I O N S  
\*\*\*\*\*

TITLE	/	(RE)SET/	OBS/TYPE	/	A/	B/	SEED
Flatford		0.000*****	UNIFORM		0.000	1.000	33052265

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
tale		0.000*****	9.420&-0062.569&-0057.510&-009-2.27&-0131.200&-004				

CELL/LOWER LIM/	N/	FREQ/	CUM :	
0 -INFINITY	92	0.00	0.00	I-----
1 0.000*****		0.93	92.53	I*****
26.250&-005873635		0.07	100.00	I**

\*\*REST OF TABLE EMPTY\*\*

I-----

S U M M A R Y

TITLE / (RE)SET/ OBS/ AVERAGE/EST.ST.DV/ STD.ERR./ MINIMUM/  
 MAXIMUM  
 video1 0.00068945675.898&-0061.615&-0056.152&-009-2.27&-0137.680&-  
 005

CELL/LOWER LIM/	N/	FREQ/	CUM :	
0 -INFINITY	15	0.00	0.00	I-----
1 0.000*****		0.97	97.31	I.
26.250&-005185686		0.03	100.00	I*****

\*\*REST OF TABLE EMPTY\*\*

I-----

S U M M A R Y

TITLE / (RE)SET/ OBS/ AVERAGE/EST.ST.DV/ STD.ERR./ MINIMUM/  
 MAXIMUM  
 video2 0.00069205141.080&-0041.098&-0044.174&-008-2.27&-0132.236&-  
 003

CELL/LOWER LIM/	N/	FREQ/	CUM :	
0 -INFINITY	3	0.00	0.00	I-----
1 0.000*****		0.48	48.40	I.
26.250&-005*****		0.15	63.04	I*****
31.250&-004956176		0.14	76.86	I*****
41.875&-004679939		0.10	86.68	I*****
52.500&-004509781		0.07	94.05	I*****
63.125&-004277526		0.04	98.06	I**
73.750&-004 96771		0.01	99.46	I*
84.375&-004 20804		0.00	99.76	I.
95.000&-004 10779		0.00	99.91	I.
105.625&-004 5038		0.00	99.98	I.
116.250&-004 673		0.00	99.99	I.
126.875&-004 154		0.00	100.00	I.
137.500&-004 81		0.00	100.00	I.
148.125&-004 49		0.00	100.00	I.
158.750&-004 30		0.00	100.00	I.
169.375&-004 21		0.00	100.00	I.
171.000&-003 73		0.00	100.00	I.

I-----

S U M M A R Y

TITLE / (RE)SET/ OBS/ AVERAGE/EST.ST.DV/ STD.ERR./ MINIMUM/  
 MAXIMUM  
 data1 0.000\*\*\*\*\*5.117&-0042.998&-0034.873&-007-1.14&-013  
 0.112

CELL/LOWER LIM/	N/	FREQ/	CUM :	
0 -INFINITY	1	0.00	0.00	I-----
1 0.000*****		0.20	20.39	I.
23.448&-005*****		0.08	28.82	I*****
36.897&-005*****		0.08	36.86	I*****
41.034&-004*****		0.08	44.50	I*****
51.379&-004*****		0.07	51.43	I*****
61.724&-004*****		0.06	57.92	I*****
72.069&-004*****		0.06	64.11	I*****
82.414&-004*****		0.05	69.38	I*****
92.759&-004*****		0.05	73.98	I*****
103.103&-004*****		0.04	78.14	I*****
113.448&-004*****		0.04	81.68	I*****
123.793&-004*****		0.03	84.43	I****
134.138&-004807797		0.02	86.56	I***
144.483&-004636198		0.02	88.24	I**
154.828&-004509447		0.01	89.59	I**
165.172&-004417379		0.01	90.69	I**
175.517&-004346636		0.01	91.61	I*
185.862&-004291013		0.01	92.38	I*
196.207&-004247259		0.01	93.03	I*
206.552&-004210660		0.01	93.59	I*
216.897&-004182159		0.00	94.07	I*
227.241&-004157972		0.00	94.49	I*
237.586&-004136676		0.00	94.85	I*
247.931&-004118660		0.00	95.16	I.
258.276&-004103270		0.00	95.43	I.
268.621&-004 90833		0.00	95.67	I.
278.966&-004 79782		0.00	95.88	I.
289.310&-004 70382		0.00	96.07	I.
299.655&-004 62601		0.00	96.24	I.
301.000&-003*****		0.04	100.00	I*****
				I-----

S U M M A R Y

TITLE / (RE)SET/ OBS/ AVERAGE/EST.ST.DV/ STD.ERR./ MINIMUM/  
 MAXIMUM  
 data2 0.00042245599.874&-0035.258&-0022.558&-005-5.68&-014  
 1.227

CELL/LOWER LIM/	N/	FREQ/	CUM :	
0 -INFINITY	1	0.00	0.00	I-----
1 0.000*****		0.50	49.64	I.
23.333&-004743170		0.18	67.23	I*****
36.667&-004331453		0.08	75.08	I*****
41.000&-003*****		0.25	100.00	I*****
				I-----

B I N S  
\*\*\*\*\*

TITLE	/	(RE)SET/	OBS/INIT/	MAX/	NOW/	AV. FREE/	AV. WAIT/QMAX
SYNC		0.000*****	0 1422		1	12.3268.335&-006	1

W A I T Q U E U E S  
\*\*\*\*\*

TITLE	/	(RE)SET/	OBS/	QMAX/	QNOW/	Q AVERAGE/ZEROS/	AV. WAIT
pakker		0.000*****	1	0		0.000*****	0.000
pakker	*	0.000*****	6555	4409	3849.120	7	0.285
EF		0.000*****	1	0		0.000*****	0.000
EF	*	0.000*****	5	0		0.349342217	9.387&-005
AF1		0.0006920514	1	0		0.000*****	0.000
AF1	*	0.0006920514	6	1		0.309 12776	2.234&-004
AF2		0.000*****	1	0		0.000*****	0.000
AF2	*	0.000*****	949	0		4.162 23347	5.498&-004
BE		0.0004224559	1	0		0.000*****	0.000
BE	*	0.0004224559	975	1		8.374 5023	9.912&-003

8 garbage collection(s) in 19.4 seconds.

**Utskrift av "non-preemption med 80 prosent belastning:**

CLOCK TIME = 5.000&-002

```
*****
*
*           T R A C I N G   S W I T C H E D   O F F
*
*****
```

Non-preemptiv simulering!

```
simtid=                5000.00
seedverdi=             10000
tot_packets=           68345017
amount_packets=        399396628172.98
amount_voice=          79914701190.42
amount_video1=         79534206152.47
amount_video2=         79834795689.28
```

amount\_data1= 144156946300.16

amount\_data2= 15955978840.64

Wasted service time:

EF : 0.00000000  
 AF1: 0.00000000  
 AF2: 0.00000000  
 BE : 0.00000000

CLOCK TIME = 5000.050

\*\*\*\*\*  
 \*  
 \* R E P O R T \*  
 \*  
 \*\*\*\*\*

D I S T R I B U T I O N S

\*\*\*\*\*

TITLE	/	(RE)SET/	OBS/TYPE	/	A/	B/	SEED
Flatford		0.000*****	UNIFORM		0.000	1.000	33052265

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
tale		0.000*****	2.869&-0053.644&-0051.065&-008-2.27&-0132.397&-004				

CELL/LOWER LIM/	N/	FREQ/	CUM :	
0 -INFINITY	36	0.00	0.00	I-----
1 0.000*****		0.83	83.01	I.
26.250&-005*****		0.15	98.18	I*****
31.250&-004186223		0.02	99.77	I*
41.875&-004 27183		0.00	100.00	I.

\*\*REST OF TABLE EMPTY\*\*

I-----

S U M M A R Y

TITLE	/	(RE)SET/	OBS/	AVERAGE/EST.ST.DV/	STD.ERR./	MINIMUM/	MAXIMUM
videol		0.00068945672.472&-0053.088&-0051.176&-008-2.27&-0131.967&-004					

CELL/LOWER LIM/	N/	FREQ/	CUM :
-----------------	----	-------	-------



CELL/LOWER LIM/	N/	FREQ/	CUM :	
0 -INFINITY	7	0.00	0.00	I----- I.
1 0.000*****		0.87	86.82	I*****
26.250&-005862657		0.13	99.34	I****
31.250&-004 44983		0.01	99.99	I.
41.875&-004 733		0.00	100.00	I.

\*\*REST OF TABLE EMPTY\*\*

I-----

S U M M A R Y

TITLE	/	(RE)SET/	OBS/	AVERAGE/EST.ST.DV/	STD.ERR./	MINIMUM/	MAXIMUM
video2		0.00069205143	246&-0054	513&-0051	716&-008	-2.27&-0132	.612&-004

CELL/LOWER LIM/	N/	FREQ/	CUM :	
0 -INFINITY	7	0.00	0.00	I----- I.
1 0.000*****		0.82	81.54	I*****
26.250&-005863133		0.12	94.02	I****
31.250&-004322225		0.05	98.67	I**
41.875&-004 91838		0.01	100.00	I.
52.500&-004 22		0.00	100.00	I.

\*\*REST OF TABLE EMPTY\*\*

I-----

S U M M A R Y

TITLE	/	(RE)SET/	OBS/	AVERAGE/EST.ST.DV/	STD.ERR./	MINIMUM/	MAXIMUM
data1		0.000*****	1.059&-0041	112&-0041	807&-008	-1.14&-0135	.120&-003

CELL/LOWER LIM/	N/	FREQ/	CUM :	
0 -INFINITY	2	0.00	0.00	I----- I.
1 0.000*****		0.36	35.72	I*****
23.448&-005*****		0.13	48.30	I*****
36.897&-005*****		0.11	58.81	I*****
41.034&-004*****		0.09	67.89	I*****
51.379&-004*****		0.08	75.57	I*****
61.724&-004*****		0.07	82.17	I*****
72.069&-004*****		0.05	87.39	I****
82.414&-004*****		0.04	91.39	I***
92.759&-004*****		0.03	94.68	I***
103.103&-004814446		0.02	96.84	I**
113.448&-004452543		0.01	98.03	I*
123.793&-004239411		0.01	98.66	I*
134.138&-004147333		0.00	99.05	I.
144.483&-004 99700		0.00	99.32	I.
154.828&-004 70728		0.00	99.50	I.
165.172&-004 50062		0.00	99.64	I.
175.517&-004 36159		0.00	99.73	I.
185.862&-004 26062		0.00	99.80	I.

196.207&-004	19242	0.00	99.85	I.
206.552&-004	13687	0.00	99.89	I.
216.897&-004	10026	0.00	99.91	I.
227.241&-004	7350	0.00	99.93	I.
237.586&-004	5483	0.00	99.95	I.
247.931&-004	4202	0.00	99.96	I.
258.276&-004	3217	0.00	99.97	I.
268.621&-004	2551	0.00	99.97	I.
278.966&-004	1986	0.00	99.98	I.
289.310&-004	1479	0.00	99.98	I.
299.655&-004	1176	0.00	99.99	I.
301.000&-003	5269	0.00	100.00	I.

I-----

S U M M A R Y

TITLE / (RE)SET/ OBS/ AVERAGE/EST.ST.DV/ STD.ERR./ MINIMUM/  
 MAXIMUM  
 data2 0.00042245592.441&-0047.365&-0043.583&-007-5.68&-0148.037&-  
 002

CELL/LOWER LIM/	N/	FREQ/	CUM :	
0 -INFINITY	1	0.00	0.00	I-----
1 0.000*****		0.78	77.56	I.
23.333&-004625888		0.15	92.37	I*****
36.667&-004167205		0.04	96.33	I*****
41.000&-003155087		0.04	100.00	I*

I-----

B I N S  
 \*\*\*\*\*

TITLE / (RE)SET/ OBS/INIT/ MAX/ NOW/ AV. FREE/ AV. WAIT/QMAX  
 SYNC 0.000\*\*\*\*\* 0 82 1 1.1541.489&-005 1

W A I T Q U E U E S  
 \*\*\*\*\*

TITLE	/	(RE)SET/	OBS/	QMAX/	QNOW/	Q AVERAGE/ZEROS/	AV. WAIT
pakker		0.000*****	1	0		0.000*****	0.000
pakker	*	0.000*****	6555	4409	3860.305	5	0.286
EF		0.000*****	1	0		0.000*****	0.000
EF	*	0.000*****	5	0	0.420141605	1.130&-004	
AF1		0.0006920514	1	0		0.000*****	0.000
AF1	*	0.0006920514	3	1	0.205 17148	1.478&-004	
AF2		0.000*****	1	0		0.000*****	0.000
AF2	*	0.000*****	68	0	1.090 35972	1.440&-004	
BE		0.0004224559	1	0		0.000*****	0.000
BE	*	0.0004224559	73	1	0.238 7968	2.818&-004	

8 garbage collection(s) in 19.5 seconds.

**Utskrift av "preemption" med 90 prosent belastning:**

```

                                CLOCK TIME = 5.000&-002
*****
*
*           T R A C I N G   S W I T C H E D   O F F
*
*****

```

Preemptiv simulering!

```

simtid=                1000.00
seedverdi=             3000

tot_packets=           15381020

amount_packets=        90121480475.57

amount_voice=          18044952312.20

amount_video1=         18010867079.22

amount_video2=         18183262792.42

amount_data1=          32319662976.10

amount_data2=          3562735315.63

```

Wasted service time:

```

EF :           0.00000000
AF1:           53.11864890
AF2:           42.06967082
BE :           5.43300484

```

```

                                CLOCK TIME = 1000.050
*****
*
*           R E P O R T
*
*****

```

```

                                D I S T R I B U T I O N S
*****

TITLE      /   (RE)SET/   OBS/TYPE      /           A/           B/           SEED
Flatford   /   0.0002283781 UNIFORM      /           0.000           1.000  43465453

```

```

                                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  
 tale 0.00026413151.067&-0052.710&-0051.668&-008-1.14&-0131.200&-  
 004

CELL/LOWER LIM/ N/ FREQ/ CUM :  
 I-----  
 0 -INFINITY 49 0.00 0.00 I.  
 1 0.000\*\*\*\*\* 0.92 91.53 I\*\*\*\*\*  
 26.250&-005223718 0.08 100.00 I\*\*\*  
 \*\*REST OF TABLE EMPTY\*\*  
 I-----

S U M M A R Y

TITLE / (RE)SET/ OBS/ AVERAGE/EST.ST.DV/ STD.ERR./ MINIMUM/  
 MAXIMUM  
 videol 0.00015615196.645&-0061.700&-0051.361&-008-1.14&-0137.680&-  
 005

CELL/LOWER LIM/ N/ FREQ/ CUM :  
 I-----  
 0 -INFINITY 16 0.00 0.00 I.  
 1 0.000\*\*\*\*\* 0.97 96.96 I\*\*\*\*\*  
 26.250&-005 47434 0.03 100.00 I\*  
 \*\*REST OF TABLE EMPTY\*\*  
 I-----

S U M M A R Y

TITLE / (RE)SET/ OBS/ AVERAGE/EST.ST.DV/ STD.ERR./ MINIMUM/  
 MAXIMUM  
 video2 0.00015755711.342&-0041.243&-0049.903&-008 0.0002.849&-  
 003

CELL/LOWER LIM/ N/ FREQ/ CUM :  
 I-----  
 0 -INFINITY 0 0.00 0.00 I  
 1 0.000629667 0.40 39.96 I\*\*\*\*\*  
 26.250&-005244155 0.15 55.46 I\*\*\*\*\*  
 31.250&-004232225 0.15 70.20 I\*\*\*\*\*  
 41.875&-004179256 0.11 81.58 I\*\*\*\*\*  
 52.500&-004140501 0.09 90.49 I\*\*\*\*\*  
 63.125&-004 83771 0.05 95.81 I\*\*\*\*  
 73.750&-004 30240 0.02 97.73 I\*  
 84.375&-004 16027 0.01 98.75 I\*  
 95.000&-004 12114 0.01 99.52 I\*  
 105.625&-004 5824 0.00 99.89 I.  
 116.250&-004 715 0.00 99.93 I.  
 126.875&-004 325 0.00 99.95 I.  
 137.500&-004 196 0.00 99.96 I.  
 148.125&-004 143 0.00 99.97 I.  
 158.750&-004 90 0.00 99.98 I.  
 169.375&-004 67 0.00 99.98 I.  
 171.000&-003 255 0.00 100.00 I.

I-----

S U M M A R Y

TITLE	/	(RE)SET/	OBS/	AVERAGE/EST.ST.DV/	STD.ERR./	MINIMUM/
MAXIMUM						
data1		0.00084819887	149	-002	0.1234.208	-005-8.88
0.824						-016

CELL/LOWER LIM/	N/	FREQ/	CUM :	
				I-----
0 -INFINITY	1	0.00	0.00	I.
1 0.000297109		0.04	3.50	I**
23.448&-005202420		0.02	5.89	I*
36.897&-005183746		0.02	8.06	I*
41.034&-004193033		0.02	10.33	I*
51.379&-004188278		0.02	12.55	I*
61.724&-004184658		0.02	14.73	I*
72.069&-004187026		0.02	16.93	I*
82.414&-004178191		0.02	19.03	I*
92.759&-004164868		0.02	20.98	I*
103.103&-004158444		0.02	22.85	I*
113.448&-004149919		0.02	24.61	I*
123.793&-004131894		0.02	26.17	I*
134.138&-004115382		0.01	27.53	I*
144.483&-004101593		0.01	28.73	I*
154.828&-004 89520		0.01	29.78	I*
165.172&-004 79531		0.01	30.72	I.
175.517&-004 71655		0.01	31.56	I.
185.862&-004 64304		0.01	32.32	I.
196.207&-004 58271		0.01	33.01	I.
206.552&-004 53053		0.01	33.63	I.
216.897&-004 48357		0.01	34.20	I.
227.241&-004 44381		0.01	34.73	I.
237.586&-004 40887		0.00	35.21	I.
247.931&-004 37841		0.00	35.66	I.
258.276&-004 34908		0.00	36.07	I.
268.621&-004 32092		0.00	36.45	I.
278.966&-004 29657		0.00	36.80	I.
289.310&-004 27798		0.00	37.12	I.
299.655&-004 25682		0.00	37.43	I.
301.000&-003*****		0.63	100.00	I*****
				I-----

S U M M A R Y

TITLE	/	(RE)SET/	OBS/	AVERAGE/EST.ST.DV/	STD.ERR./	MINIMUM/
MAXIMUM						
data2		0.000	885468	33.696	13.2921.413	-002
73.713						0.000

CELL/LOWER LIM/	N/	FREQ/	CUM :	
				I-----
0 -INFINITY	0	0.00	0.00	I

```

1      0.000  2036   0.00   0.23   I.
23.333&-004  847   0.00   0.33   I.
36.667&-004  417   0.00   0.37   I.
41.000&-003882168  1.00  100.00  I*****
I-----

```

B I N S

\*\*\*\*\*

```

TITLE      /   (RE)SET/   OBS/INIT/  MAX/ NOW/  AV. FREE/  AV. WAIT/QMAX
SYNC              0.000*****   07142157919 32169.7743.043&-008  1

```

W A I T Q U E U E S

\*\*\*\*\*

```

TITLE      /   (RE)SET/   OBS/ QMAX/ QNOW/ Q AVERAGE/ZEROS/  AV. WAIT
pakker          0.000*****   1    0    0.000*****   0.000
pakker          *   0.000***** 27632 14372  9768.186   291   0.645
EF              0.0004202834   1    0    0.000*****   0.000
EF              *   0.0004202834   4    0    0.399 75513 9.497&-005
AF1             0.0001575571   1    0    0.000*****   0.000
AF1             *   0.0001575571   5    0    0.393   905 2.496&-004
AF2             0.0008481988   1    0    0.000*****   0.000
AF2             *   0.0008481988  7692  0    606.671   235 7.153&-002
BE              0.000 885468   1    0    0.000885468   0.000
BE              *   0.000 885468 71323 57920 31563.310   3    33.696

```

1 garbage collection(s) in 2.5 seconds.

**Utskrift av "non-preemption" med 90 prosent belastning:**

CLOCK TIME = 5.000&-002

```

*****
*
*           T R A C I N G   S W I T C H E D   O F F
*
*****

```

Non-preemptiv simulering!

```

simtid=                1000.00

seedverdi=              500

tot_packets=           15414151

amount_packets=        90217702436.60

amount_voice=          18041087013.00

amount_video1=         18112262182.09
amount_video2=         18037435072.54

amount_data1=          32418952561.07

amount_data2=          3607965607.88

```

Wasted service time:

EF : 0.00000000  
 AF1: 0.00000000  
 AF2: 0.00000000  
 BE : 0.00000000

CLOCK TIME = 1000.050

\*\*\*\*\*  
 \*  
 \* R E P O R T \*  
 \*  
 \*\*\*\*\*

D I S T R I B U T I O N S

\*\*\*\*\*

TITLE	/	(RE)SET/	OBS/TYPE	/	A/	B/	SEED
Flatford		0.0002283079	UNIFORM		0.000	1.000	18427500

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						
tale		0.00026406093		3.441&-0053.765&-0052.317&-008	1.14&-0132.396&-	004

CELL/LOWER LIM/	N/	FREQ/	CUM :
0 -INFINITY	8	0.00	0.00
1 0.000*****		0.80	80.01
26.250&-005459416		0.17	97.40
31.250&-004 60711		0.02	99.70
41.875&-004 7819		0.00	100.00

\*\*REST OF TABLE EMPTY\*\*

I-----

S U M M A R Y

TITLE	/	(RE)SET/	OBS/	AVERAGE/EST.ST.DV/	STD.ERR./	MINIMUM/
MAXIMUM						
videol		0.00015697322		2.997&-0053.154&-0052.518&-008	1.14&-0131.967&-	004

CELL/LOWER LIM/	N/	FREQ/	CUM :	
0 -INFINITY	5	0.00	0.00	I----- I.
1 0.000*****		0.84	84.15	I*****
26.250&-005235422		0.15	99.14	I*****
31.250&-004 13238		0.01	99.99	I.
41.875&-004 207		0.00	100.00	I.

\*\*REST OF TABLE EMPTY\*\*

I-----

S U M M A R Y

TITLE	/	(RE)SET/	OBS/	AVERAGE/EST.ST.DV/	STD.ERR./	MINIMUM/ MAXIMUM
video2		0.00015637764		.219&-0055.045&-0054.034&-008-1.14&-0132.623&-004		

CELL/LOWER LIM/	N/	FREQ/	CUM :	
0 -INFINITY	5	0.00	0.00	I----- I.
1 0.000*****		0.76	75.86	I*****
26.250&-005238428		0.15	91.11	I*****
31.250&-004101257		0.06	97.58	I***
41.875&-004 37799		0.02	100.00	I*
52.500&-004 17		0.00	100.00	I.

\*\*REST OF TABLE EMPTY\*\*

I-----

S U M M A R Y

TITLE	/	(RE)SET/	OBS/	AVERAGE/EST.ST.DV/	STD.ERR./	MINIMUM/ MAXIMUM
data1		0.00085071096		.748&-0045.395&-0031.850&-006-1.14&-0130.134		

CELL/LOWER LIM/	N/	FREQ/	CUM :	
0 -INFINITY	2	0.00	0.00	I----- I.
1 0.000*****		0.21	20.85	I*****
23.448&-005803848		0.09	30.30	I*****
36.897&-005728458		0.09	38.86	I*****
41.034&-004691732		0.08	46.99	I*****
51.379&-004644534		0.08	54.57	I*****
61.724&-004606460		0.07	61.70	I*****
72.069&-004537269		0.06	68.01	I*****
82.414&-004466741		0.05	73.50	I*****
92.759&-004420617		0.05	78.44	I*****
103.103&-004337077		0.04	82.40	I*****
113.448&-004246112		0.03	85.30	I****



---

123.793&-004176003	0.02	87.37	I***
134.138&-004139198	0.02	89.00	I**
144.483&-004111623	0.01	90.32	I**
154.828&-004 91721	0.01	91.39	I**
165.172&-004 76400	0.01	92.29	I*
175.517&-004 63955	0.01	93.04	I*
185.862&-004 53544	0.01	93.67	I*
196.207&-004 45231	0.01	94.20	I*
206.552&-004 37930	0.00	94.65	I*
216.897&-004 31697	0.00	95.02	I*
227.241&-004 26899	0.00	95.34	I.
237.586&-004 22723	0.00	95.61	I.
247.931&-004 19189	0.00	95.83	I.
258.276&-004 16386	0.00	96.02	I.
268.621&-004 14124	0.00	96.19	I.
278.966&-004 12295	0.00	96.33	I.
289.310&-004 10784	0.00	96.46	I.
299.655&-004 9369	0.00	96.57	I.
301.000&-003291662	0.03	100.00	I*****

I-----

S U M M A R Y

TITLE	/	(RE)SET/	OBS/	AVERAGE/EST.ST.DV/	STD.ERR./	MINIMUM/	MAXIMUM
data2		0.000	9550531	1.682&-0029.164&-0029.378&-005-1.14&-013		2.016	

CELL/LOWER LIM/	N/	FREQ/	CUM :
0 -INFINITY	1	0.00	0.00
1 0.000464430		0.49	48.63
23.333&-004151429		0.16	64.48
36.667&-004 70543		0.07	71.87
41.000&-003268650		0.28	100.00

I-----

B I N S  
\*\*\*\*\*

TITLE	/	(RE)SET/	OBS/INIT/	MAX/	NOW/	AV. FREE/	AV. WAIT/QMAX
SYNC		0.000*****	0 3392	1	22.0036.436&-006	1	

W A I T Q U E U E S  
\*\*\*\*\*

TITLE	/	(RE)SET/	OBS/	QMAX/	QNOW/	Q AVERAGE/ZEROS/	AV. WAIT
pakker		0.000*****	1	0	0.000*****	0.000	
pakker	*	0.000*****	9128	7073	6563.520	14	0.431
EF		0.0004210341	1	0	0.000*****	0.000	
EF	*	0.0004210341	5	0	0.499 20223	1.186&-004	
AF1		0.0001563776	1	0	0.000*****	0.000	
AF1	*	0.0001563776	3	0	0.246 2374	1.575&-004	
AF2		0.0008507109	1	0	0.000*****	0.000	

---

AF2	*	0.0008507109	1554	1	6.064	4090	7.129&-004
BE		0.000 955053	1	0	0.000955053		0.000
BE	*	0.000 955053	2372	1	16.095	878	1.685&-002

1 garbage collection(s) in 2.4 seconds

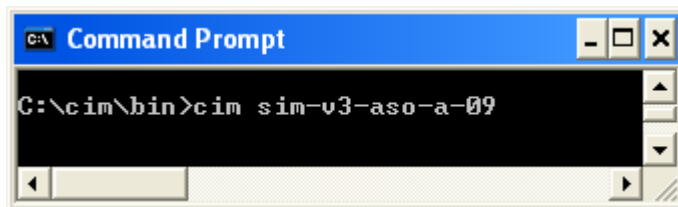
### 9.3 Vedlagt CD

### 9.4 Oppsett og kjøring av programmet

1. En PC med minimum 1,5 GB RAM.
2. Operativsystemer (linux eller windows xp). Windows XP med "Comand Prompt" har blitt brukt, men programmet har blitt testet på en linux maskin uten problemer.
3. Man kan installere Simula/Cim fra faget TTM4110 (Dependability and performance with discrete event simulation) under Simula/Cim.
4. Se følgende skjermbilder for å kompilere og kjøre programmet:

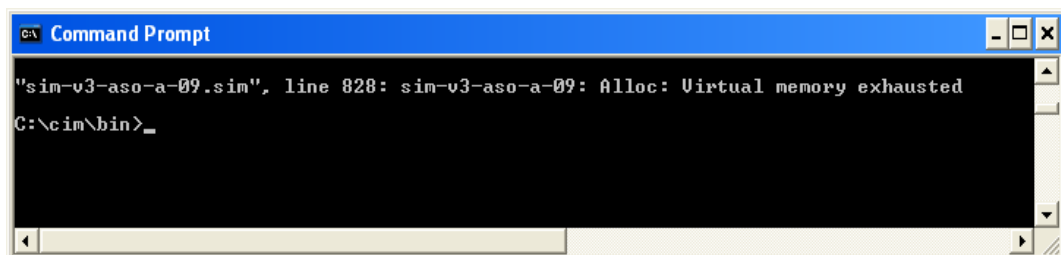
Kompilering:

sim-v3-aso-a-09 er navnet programmet.



```
C:\cim\bin>cim sim-v3-aso-a-09
```

Det er nesten vanlig å få problemer med memory (Typisk feilmelding "Virtual memory exhausted") som følgende.

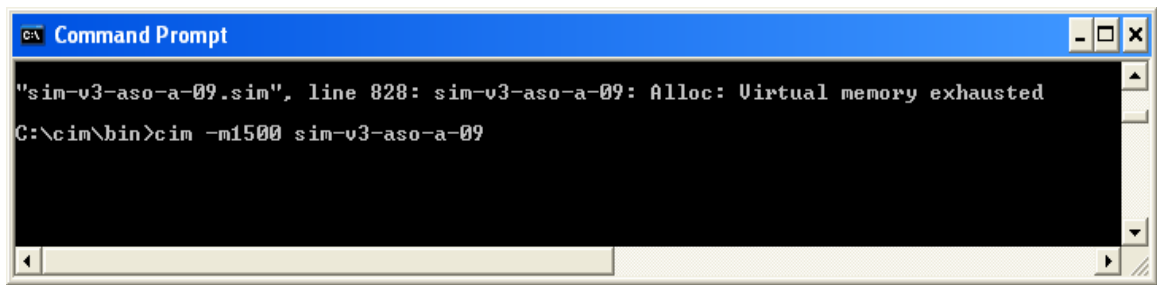


```
"sim-v3-aso-a-09.sim", line 828: sim-v3-aso-a-09: Alloc: Virtual memory exhausted
C:\cim\bin>_
```

Man kompilerer programmet med følgende opsjon:

```
cim -m1500 sim-v3-aso-a-09
```

Følgende skjermbilde viser kompilering med oppnevnte opsjonen:

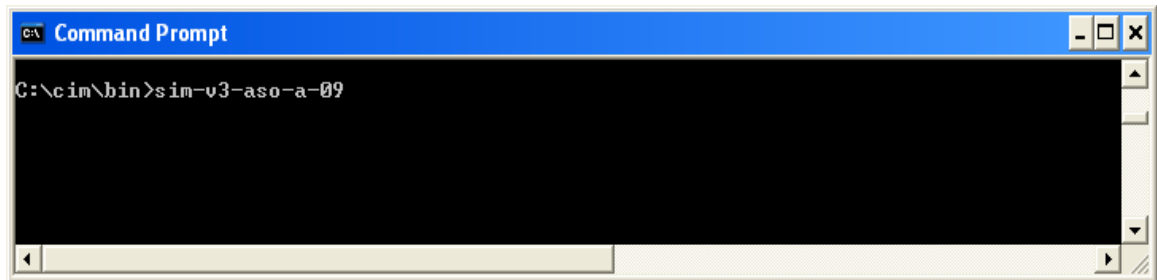


```
C:\ Command Prompt

"sim-v3-aso-a-09.sim", line 828: sim-v3-aso-a-09: Alloc: Virtual memory exhausted
C:\cim\bin>cim -m1500 sim-v3-aso-a-09
```

For å gå videre

Deretter kjør programmet som følgende skjermbilde:



```
C:\ Command Prompt

C:\cim\bin>sim-v3-aso-a-09
```