

TEKNILLINEN KORKEAKOULU
Teletekniikan laboratorio
S-38.128 Teletekniikan erikoistyö

14.9.2000

Optisen pakettimultiplekserin suorituskykyanalyysi

Riikka Susitaival
49951A

Ohjaaja Samuli Aalto

Sisällysluettelo

Sisällysluettelo	2
Tiivistelmä	3
Lyhenne- ja käsiteluettelo	4
1. Johdanto	5
2. Optinen pakettimultiplekseri	6
2.1 Viivelinjat	6
2.2 Matemaattinen mallinnus	7
2.2 Analyttinen laskeminen	9
2.3 Optimaalinen viivelinjakonfiguraatio	9
3. Konfiguraatiot, kun $F=10$	10
3.1 Konfiguraatioiden ja kuorman valitseminen	10
3.2 $F=10$, m vaihtelee	11
3.3 Menetyssuhteen muuttuminen erilaisilla kuormilla	13
3.4 Hyvät ja huonot konfiguraatiot	16
4. Menetyssuhteen analysointi simuloiden	17
5. Laskeminen Mathematicalla eri parametrein	19
6. Simulaattorin suorituskyvystä	20
7. Yhteenveto	22
Lähdeluettelo	23
Liitteet	24

Tiivistelmä

Täysoptisen pakettiverkon yksi komponentti on optinen pakettimultiplekseri. Puskurointi optisessa multiplekserissa voidaan toteuttaa viivelinjoja käyttämällä. Työn tarkoituksena on analysoida tällaisen puskurin suorituskykyä erilaisin matemaattisin mallein ja parametrein.

Työssä on keskitytty pakettien menetyssuhteeseen suorituskyvyn mittaajana. Menetyssuhteen analyttiselle laskemiselle on toteutettu Mathematica-paketti. Toisena työvälineenä on Jaakko Tiasen erikoistyössään toteuttama simulaattori.

Viivelinjojen pituuksien ja määrän, eli konfiguraation suunnittelulla ja mitoituksella, voidaan saavuttaa huomattavaa etua menetyssuhteessa. Tutkitaan, minkälainen konfiguraatio on optimaalinen.

Kun viivelinjat ovat jaksollisia eli pituudet kasvavat tasaisesti, esimerkiksi pituudet 1, 2, 3 ja 4 tai 1, 3, 5 ja 7 aikayksikköä, voidaan puskurin mallintaa FIFO-puskurina ja menetyssuhde on sama kuin puskurin, jossa on sama määrä odotuspaikkoja kuin on tämän mallin viivelinjojen määrä. Tällöin ei kuitenkaan saavuteta pientä menetyssuhdetta.

Pienen menetyssuhteen saavuttamiseksi ei olla löydetty suoraviivaista ratkaisua. Kuitenkin konfiguraatio, jonka viivelinjat on valittu satunnaisesti, ja siinä on mukana sekä lyhyitä että pitkiä viivelinjoja, antaa melko varmasti pienen menetyssuhteen. Vastaavasti eräillä muilla algoritmeilla valitut konfiguraatiot ovat hyviä.

Kun viivelinjojen määrä ja maksimipituus on kiinnitetty, menetyssuhteen ero huonon, FIFO-puskurin tapaan mallinnettavan konfiguraation ja parhaan mahdollisen konfiguraation välillä 1000% luokkaa. Toisaalta erot parhaassa päässä ovat melko pienet.

Lisäksi on pohdittu analyttisen laskennan ja simulaation eroja. Analyttinen ratkaisu osoittautuu hyväksi pienillä parametreilla, koska nopeasti saadaan tarkkoja tuloksia. Simulaatio on taas parempi menetelmä, kun viivelinjojen maksimipituus kasvaa yli 10 aikayksikköä.

Lyhenne- ja käsiteluettelo

N sisääntulolinjojen lukumäärä

m viivelinjojen lukumäärä

F viivelinjojen maksimipituus eli puskurin syvyys

l_i yksittäisen viivelinjan pituus

p todennäköisyys että yhdeltä sisääntulolinjalta saapuu paketti aikavälissä

Δ aikaväli

1. Johdanto

Täysoptisen pakettiverkon toteuttamisessa yksi ratkaistava ongelma on puskurointi. Yksi tapa toteuttaa puskuri eli optinen pakettimultiplekseri on viivelinjojen käyttö. Työn teorian lähtökohtana on ollut Cadron, Graveyn ja Guillemot'n julkaisu *Performance evaluation of an optical transparent packet switch* [1].

Viivelinjojen ideana on johtaa puskuroitava liikenne eri mittaisille linjoille, jotta ne saapuisivat palveltavaksi eri ajan hetkinä. Viivelinjojen pituuksien suunnittelulla voidaan kuitenkin saada erilaisia tuloksia mm. pakettien menetyssuhteessa.

Työn tarkoituksena on tutkia viivelinjojen pituuksien valintaa siten, että optimi menetyssuhteessa ja viiveessä saavutetaan. Puskurin pituus ja viivelinjojen määrä on rajallinen.

Työkaluna on valmiiksi tehty simulaattori menetyssuhteen simulointiin. Tämä on Jaakko Tiasen erikoistyössään [2] toteuttama simulaattori. Lisäksi kehitetään analyttinen malli viivelinjojen menetyssuhteelle, jonka mukaan geneerisesti lasketaan Mathematicalla. Tämä Mathematica-ohjelma kootaan paketiksi.

Yritetään siis muodostaa matemaattisia malleja ja algoritmeja viivelinjojen suorituskyvyn analysointiin. Lisäksi työssä vertaillaan analyysin ja simulaation laskennan tarkkuuden ja ajankäytön eroja eri parametrein sekä etsitään kummankin tavan laskentarajoja erisuuruisilla parametreilla.

2. Optinen pakettimultiplekseri

2.1 Viivelinjat

Tarkastelun kohteena on aikaviipaloitu järjestelmä, jossa paketit ovat kiinteän mittaisia, kuten esimerkiksi ATM:ssä. Liikennemallina ajatellaan, että yhdessä aikavälissä (Δ) saapuvien pakettien määrä on binomijakautunut ja saapumiset ovat toisistaan riipumattomia.

Todennäköisyys että yhdeltä sisääntulolinjalta saapuu paketti yhdessä aikayksikössä on p . Kun sisääntulolinjoja on N , on todennäköisyys, että paketteja saapuu i kappaletta aikayksikössä

$$p(i) = \binom{N}{i} p^i (1-p)^{N-i}$$

Kun paketti saapuu puskuriin, se sijoitetaan lyhimmälle mahdolliselle viivelinjalle. Jotta kaksi pakettia ei tulisi yhtä aikaa palveltavaksi, viivelinja on mahdollinen vain, jos yhtään pakettia ei aikaisemmin olla suunniteltu palveltavaksi samalla hetkellä, kuin on kyseisen viivelinjan pituus.

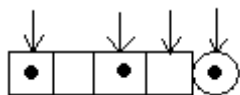
Mallia voi tarkastella eräänlaisena jonopuskurina, jossa vain tiettyihin paikkoihin on mahdollista laittaa paketteja.

Jos viivelinjat ovat pituuksiltaan 1,2..m aikaväliä, puskuria voidaan mallintaa tavallisena FIFO-jonona, koska viivelinjat täyttyvät pituusjärjestyksessä ja paketit tulevat tasaisesti palveltaviksi.

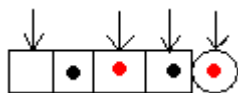
Toisaalta taas viivelinjojen pituudet voidaan valita muutenkin. Seuraavassa esimerkissä mallinnetaan tilannetta, joissa viivelinjojen pituudet ovat 1 2 3 ja 5 aikayksikköä.

Jonoissa nuolet osoittavat viivelinjoja, mustat täplät vanhoja paketteja ja punaiset uusia paketteja. Kutakin pakettia liikutetaan oikealle kun aika etenee.

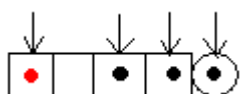
Esimerkki 1.



Kuva 1. Alkutilanne, puskurissa kolme pakettia



Kuva 2. Vanhoja paketteja siirretty oikealle, yksi palvelu ja kaksi uutta pakettia sijoitettu viivelinjoille 1 ja 3.



Kuva 3. Paketteja siirretty oikealle ja yksi saapunut paketti sijoitettu viivelinjalle 5.

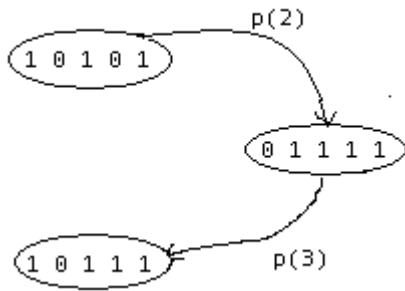
Jos paketteja saapuu samassa aikavälissä enemmän kuin on vapaana mahdollisia viivelinjoja, ylimääräiset paketit menetetään.

2.2 Matemaattinen mallinnus

Edellä kuvattu optinen pakettimultiplekseri voidaan mallintaa stationaarisena Markov-prosessina. Prosessin tulevaisuus ja menneisyys ovat toisistaan riippumattomia, kun annetaan prosessin tila jollakin tietyllä hetkellä. Prosessi on sekä diskreettiaikainen että -tilainen.

Prosessin tilaa kuvaa vektori $\underline{X} = (X_1, \dots, X_F)$, jossa X_i kertoo, onko nykyhetkessä i :n aikayksikön päähän allokoitu pakettia (1) vai ei (0).

Tutkitaan prosessin tilojen muuttumista eri todennäköisyyksillä $p(i)$. Edellisen esimerkin siirtymät ovat kuvassa 4. Ykköset kuvaavat varattuja viivelinjoja ja nollat vapaita ja mahdollisia. Oikealla oleva paketti pääsee palveluun ensimmäisenä.



Kuva 4. Tilasiirtymät, kun ensimmäiseen tilaan saapuu kaksi pakettia aikayksikössä ja toiseen tilaan kolme.

Tutkitaan siirtymät yhdestä tilasta uuteen tilaan kaikilla saapumistodennäköisyyksillä, aina siihen asti että yhdessä aikayksikössä saapuu N kpl paketteja. Näin saadaan siirtymätodennäköisyydet $p_{i,j}$ tilasta i tilaan j . Varsinaiseen tilajoukkoon Ω otetaan mukaan kaikki ne tilat j , joihin on pääsy tilasta $(0,0,\dots,0)$.

Tämä äärellinen Markov-prosessi täyttää globaalit tasapainoehdot ja tilatodennäköisyys saadaan kaavasta

$$\pi_j = \sum_i \pi_i p_{i,j}.$$

Tilayhtälöt voidaan ratkaista, kun otetaan huomioon normeerausyhtälö

$$\sum_j \pi_j = 1.$$

Globaaleista tasapainoyhtälöistä ja normeerausyhtälöstä saadaan lineaarinen yhtälöryhmä, jonka ratkaisuna saadaan tilatodennäköisyydet.

Olkoon A saapuvien pakettien lukumäärä yhdessä aikayksikössä. Koska saapumisprosessi on binomijakautunut, on odotusarvo saapumisten lukumäärälle

$$E(A) = Np.$$

Toisaalta olkoon S aikayksikössä palveltujen pakettien lukumäärä. Odotusarvo palveluille paketeille on

$$E(S) = \sum \pi_k, k \in \Omega_S,$$

eli se on niiden tilojen summa, joista paketti pääsee palveluun. Kuvassa 4 ykkönen oikeassa reunassa osoittaa tätä.

Pakettien menetystodennäköisyys saadaan palvelimeen saapuneiden, mutta menetettyjen ja palveluun saapuneiden ja palveltujen pakettien suhteesta eli

$$P_{loss} = \frac{E[A] - E[S]}{E[A]}.$$

2.2 Analyyttinen laskeminen

Edellisen kuvauksen mukaiseen analyttiseen malliin perustuva geneerinen ohjelma on toteutettu Mathematicalla ja koottu paketiksi.

Parametreiksi ohjelmalle annetaan puskurin syvyys ja viivelinjojen konfiguraatio eli eri pituuksien yhdistelmä listamuodossa, esimerkiksi edellisessä esimerkissä konfiguraatio olisi $\{1,2,3,5\}$. Lisäksi parametrina annetaan sisääntulolinjojen lukumäärä, ohjelmassa parametri n , ja saapumistodennäköisyys p .

Ohjelma laskee pakettien menetyssuhteen. Paketti on liitteenä 1.

2.3 Optimaalinen viivelinjakonfiguraatio

Pakettien menetyssuhteeseen vaikuttavat suoraan viivelinjojen lukumäärä eli mitä enemmän viivelinjoja on, sitä enemmän on muistipaikkoja. Yleisesti ensimmäisen viivelinjan pituudeksi tulee asettaa 1, sillä jos puskuri on tyhjä, tulevaa pakettia ei turhaan kierrätetä.

Toisaalta viivelinjojen pituuksien valinnoilla voidaan oleellisesti vaikuttaa menetyssuhteeseen. Mitä isompi on pisimmän viivelinjan pituus eli puskurin syvyys, sitä enemmän on tilaa säilyttää paketteja ja sitä parempi menetyssuhde on keskimäärin.

Kun viivelinjojen pituuksiksi valitaan järjestyksessä luvut $1,2,3,4..F$, voidaan multiplekseri mallintaa tavallisena FIFO-puskurina. Saapunut paketti sijoitetaan lyhimmälle vapaalle linjalle jonon loppuun. Pakettien järjestys säilyy samana.

Jos viivelinjat on valittu muuten kuin järjestyksessä alkupään luvuista, ei palveluun saapuvien pakettien järjestyksen säilymistä voida taata. Mutta näin saavutetaan parempi menetyssuhde, sillä hyödynnetään niitä aikavälejä, joiden pituisia viivelinjoja ei ole.

Seuraavassa on luokiteltu eräitä tapauksia, jotka palautuvat normaaliksi FIFO-puskuriksi tai muuksi yksinkertaisemmaksi malliksi. Näissä tapauksissa ei saavuteta etua konfiguraatiota muuntelemalla.

1)

Tapaukset, joissa viivelinjat ovat peräkkäisiä eli $l = \{i, i + 1, i + 2, \dots, i + (m - 1)\}$. Nämä mallinetaan FIFO-puskurin tapaan. Tällaisella konfiguraatiolla pakettien järjestys voidaan taata.

Sama menetyssuhde saavutetaan myös konfiguraatioilla lisäykset tasavälisiä eli $l = \{i, i + 2, i + 4, i + 6, \dots, i + 2 \cdot (m - 1)\}, \{i, i + 3, i + 6, i + 9, \dots, i + 3 \cdot (m - 1)\}, \dots, \{i, i + k, i + 2 \cdot k, i + 3 \cdot k, \dots, i + (m - 1) \cdot k\}$.

Tämä seuraa suoraan todistuksesta kohdassa 3.

Tässä joukossa optimaalisin konfiguraatio on $l = \{1, 2, 3, \dots, m\}$, jolloin saavutetaan lyhin viive.

2)

Tapauksessa $m=2$ menetystodennäköisyys on riippumaton konfiguraatiosta.

Tämä on seurausta siitä että

a)

kun $F = 2$, eli konfiguraatio $\{1, 2\}$, menetyssuhde lasketaan kuten tavallisen äärellisen jonopuskurin menetyssuhde, jononpituusprosessin stationaarisesta jakaumasta.

b)

Yleisesti, kun konfiguraatio on $\{1, F\}$, tarkastellaan ajan hetkille $n+1$ ja $n+F$ allokoitujen pakettien lukumäärää ajanhetkillä $n = j, j + (F - 1), j + 2 \cdot (F - 1) \dots$ missä $j = 0, 1 \dots F-2$. Tällainen stokastinen prosessi on Markov-ketju ja käyttäytyy saapumisten riippumattomuuden nojalla samoin kuin tavallista äärellistä puskuria vastaava jonotusprosessi, siis kuten kohdassa a.

3)

Olkoon konfiguraatio L , jossa viivelinjojen pituudet $l_1, l_2, l_3 \dots l_m$, viivelinjojen määrä m ja maksimipituus F . Jos $g := \text{s.y.t } \{ l_i - 1, i = 2, 3 \dots m \} > 1$, niin on olemassa saman suorituskyvyn omaava konfiguraatio L' , jossa on sama viivelinjojen määrä m , mutta maksimipituus on F/g .

Tutkitaan saapumisia ja systeemin tilaa hetkillä $g \cdot t, g \cdot t + 1, \dots, g \cdot t + (g - 1)$.

Kokonaissysteemi voidaan jakaa g :hen eri systeemiin, jotka ovat itsenäisiä ja käyttäytyvät toisistaan riippumattomasti. Koska yhteen tällaiseen osasysteemiin voidaan ajatella tulevan g :nnes osa kaikista saapumisista, saadaan

kokonaismenetyssuhteeksi $g \cdot \frac{1}{g} \cdot B_s = B_s$. Ollaan siis muodostettu saman

menetyssuhteen omaava ekvivalentti systeemi [3]. Esimerkkinä konfiguraatiot $\{1, 2, 3, 5\} \equiv \{1, 3, 5, 9\} \equiv \{1, 4, 7, 13\} \equiv \dots$, joilla on sama menetyssuhde.

3. Konfiguraatiot, kun $F=10$

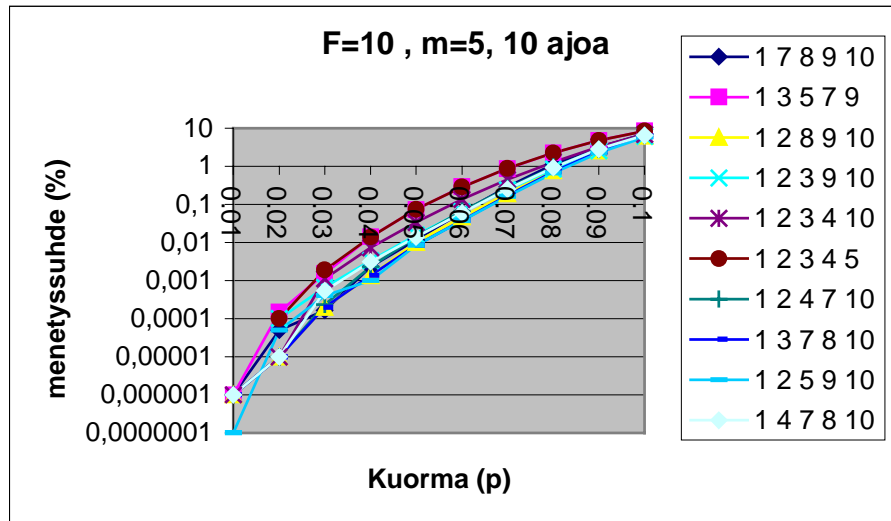
3.1 Konfiguraatioiden ja kuorman valitseminen

Tutkitaan tapausta, missä $N=10$ ja $p:n$ annetaan vaihdella. Optimaalisen viivelinjan löytämiseksi tutkitaan analyyttisesti laskemalla järjestelmällisesti kaikki konfiguraatiot, kun $F=10$ ja $m=2, 3 \dots 10$. Lyhimmälle viivelinjalle asetetaan kuitenkin pituudeksi arvo 1.

Kutakin viivelinjan pituutta kohden erilaisia konfiguraatioita on

$$\binom{F-1}{m-1} \text{ kappaletta.}$$

Kuorman odotusarvo on $E[A]=Np$. Kun p vaihtelee 0,01:stä 0,10:aan, kuorma vaihtelee 0,1:stä yhteen. Simuloidaan aluksi muutamalla eri konfiguraatiolla, jotta selviää minkälaisilla p :n arvoilla menetyssuhteen vaihtelu eri konfiguraatioiden välillä on suurin.

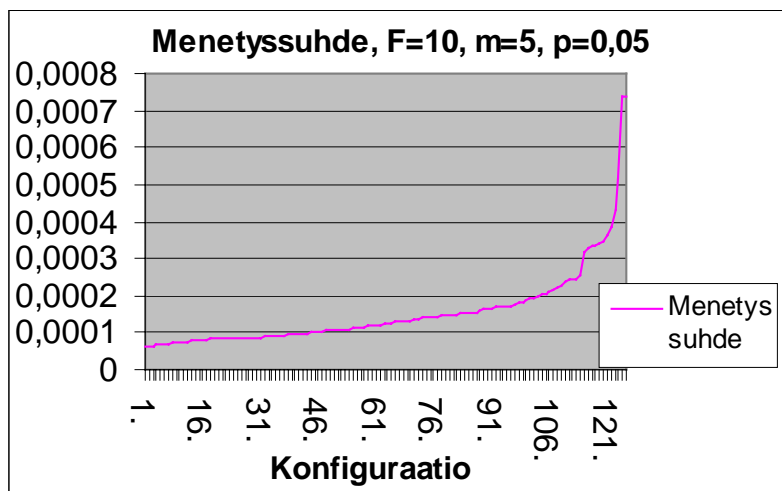


Kuva 4.

Kuvasta 4 huomataan, että logaritmisella asteikolla hajonta eri konfiguraatioiden kesken m :n arvolla 5 on suurin suunnilleen kuormalla 0,05. Kun m on suurempi, saadaan kuormaa kasvattamalla suurempi hajonta.

3.2 F=10 , m vaihtelee

Esimerkkinä otetaan tapaus $F=10$ ja $m=5$. Tulokset on järjestetty siten, että parhaan menetyssuhteen omaava konfiguraatio on listassa ensimmäisenä jne. Kuvaaajasta ilmenee menetyssuhteen kehittyminen, konfiguraatiot ja menetyssuhteet on taulukoitu liitteessä 2.



Kuva 5. Menetyssuhteet, kun konfiguraatiot paremmuusjärjestyksessä.

Huomataan, että menetyssuhde kasvaa yli kymmenkertaiseksi, kun konfiguraatiota muutetaan. Huonoimmat konfiguraatiot ovat jaksollisia ja säännöllisiä, juuri niitä jotka on esitetty kappaleessa 2. Parhaimmista taas on vaikeampi vetää vielä suoria johtopäätöksiä.

Vastaavasti on tutkittu tapaukset kun $F=10$ ja $m= 3, 4, 6, 7, 8, 9$ ja 10 . Kun m on pienempi suhteessa F :ään, on enemmän mahdollisuuksia muodostaa jaksollisia rakenteita konfiguraatioon.

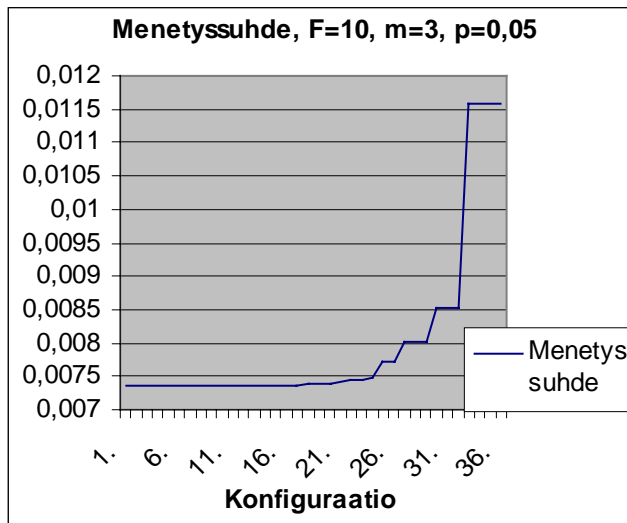
Tapauksesta $F=10$, $m=3$ ilmenee parhaiten kappaleen 2 tapaus 3, joissa viivelinjojen lisäyksen suurin yhteinen tekijä on suurempi kuin yksi. Tämä tapaukset muodostavat ryhmiä menetyssuhteen mukaisen järjestyksen loppupäähän. Näistä pitkäviivelinjaisille konfiguraatioille löytyy aina joku toinen saman menetyssuhteen omaava konfiguraatio, jolla kuitenkin lyhemmät viivelinjat. Koska menetyssuhde huononee kun F pienenee, ovat nämä konfiguraatiot huonoja.

Oheisessa listassa nämä ryhmät on tummennettu. Vastaavat ryhmät voidaan tietysti löytää myös muille konfiguraatioille, jos vain F :ää kasvatetaan. Kuvasta näkyvät nämä hyppäykset menetyssuhteessa.

$F=10, m=3$	Konfiguraatio	Menetyssuhde
1.	1 9 10	0,00736058
2.	1 8 10	0,00736059
3.	1 8 9	0,00736062
4.	1 6 10	0,00736063
5.	1 5 10	0,0073607
6.	1 6 9	0,00736076
7.	1 7 8	0,0073609
8.	1 3 10	0,00736111
9.	1 6 8	0,00736119
10.	1 4 9	0,00736139
11.	1 2 10	0,00736171
12.	1 5 8	0,00736182
13.	1 6 7	0,00736274
14.	1 4 8	0,00736315
15.	1 2 9	0,00736415
16.	1 3 8	0,00736594
17.	1 2 8	0,00737184
18.	1 5 6	0,00737506
19.	1 4 6	0,00738734
20.	1 2 7	0,0073961
21.	1 3 6	0,00741454
22.	1 4 5	0,00745755
23.	1 7 9	0,00745755
24.	1 2 6	0,00747279
25.	1 3 9	0,00771736
26.	1 2 5	0,00771736
27.	1 7 10	0,00800572
28.	1 5 7	0,00800572
29.	1 3 4	0,00800572

30.	1 2 4	0,00852364
31.	1 4 10	0,00852364
32.	1 3 7	0,00852364
33.	1 4 7	0,0115701
34.	1 3 5	0,0115701
35.	1 2 3	0,0115701
36.	1 5 9	0,0115701

Taulukko 2. F=10 ja m=3



Kuva 6. Menetyssuhde, F=10, m=3

3.3 Menetyssuhteen muuttuminen erilaisilla kuormilla

Seuraavassa taulukossa on kuvattu lisää tapausta F=10 ja m=5. Kuorma puolestaan vaihtelee arvoista p=0,03:sta 0,08:saan.

	Konfiguraatio	p=0,03	Konfiguraatio	p=0,04	Konfiguraatio	p=0,05
1.	1 4 8 9 10	1,74E-06	1 4 8 9 10	1,157E-05	1 2 7 9 10	6,08E-05
2.	1 3 8 9 10	1,74E-06	1 3 8 9 10	1,158E-05	1 3 8 9 10	6,24E-05
3.	1 6 7 9 10	1,75E-06	1 2 7 9 10	1,177E-05	1 4 8 9 10	6,26E-05
4.	1 6 8 9 10	1,76E-06	1 6 7 9 10	1,192E-05	1 2 6 9 10	6,6E-05
5.	1 2 7 9 10	1,78E-06	1 6 8 9 10	1,206E-05	1 2 6 8 9	6,82E-05
6.	1 2 6 9 10	1,80E-06	1 2 6 9 10	1,224E-05	1 6 7 9 10	6,87E-05
7.	1 5 6 8 9	1,81E-06	1 2 6 8 9	1,251E-05	1 6 8 9 10	7,05E-05
8.	1 5 6 8 10	1,81E-06	1 3 6 7 9	1,288E-05	1 2 7 8 10	7,48E-05
9.	1 3 6 7 9	1,82E-06	1 3 6 7 10	1,302E-05	1 3 6 7 10	7,53E-05
10.	1 2 6 8 9	1,83E-06	1 5 6 8 9	1,311E-05	1 3 6 7 9	7,6E-05
11.	1 3 6 7 10	1,84E-06	1 5 6 8 10	1,326E-05	1 4 6 9 10	7,61E-05
12.	1 3 6 7 8	1,85E-06	1 5 7 8 10	1,358E-05	1 5 8 9 10	7,66E-05

		p=0,06		p=0,07		p=0,08
1.	1 2 7 9 10	0,0002871	1 2 7 9 10	0,001339	1 2 7 9 10	0,00594
2.	1 3 8 9 10	0,0003164	1 2 6 9 10	0,001477	1 2 6 9 10	0,006272
3.	1 2 6 9 10	0,0003206	1 3 8 9 10	0,001560	1 2 4 9 10	0,006529
4.	1 4 8 9 10	0,0003232	1 2 6 8 9	0,001619	1 2 5 9 10	0,006603
5.	1 2 6 8 9	0,0003405	1 2 7 8 10	0,001620	1 2 7 8 10	0,006799
6.	1 2 7 8 10	0,0003561	1 4 8 9 10	0,001631	1 2 6 8 9	0,006916
7.	1 3 7 8 10	0,0003659	1 2 4 9 10	0,001658	1 3 8 9 10	0,06918
8.	1 4 6 9 10	0,0003732	1 2 5 9 10	0,001658	1 3 7 8 10	0,007182
9.	1 6 7 9 10	0,0003821	1 3 7 8 10	0,001699	1 4 6 9 10	0,007196
10.	1 2 5 9 10	0,0003839	1 4 6 9 10	0,001730	1 4 8 9 10	0,007268
11.	1 2 4 9 10	0,0003896	1 3 6 9 10	0,001853	1 3 4 9 10	0,007279
12.	1 3 6 7 10	0,0003969	1 3 4 9 10	0,001863	1 3 6 9 10	0,007316

Taulukko 3. F=10, m=5, parhaat konfiguraatiot eri kuormilla

Oheisesta taulukosta nähdään, että paras konfiguraatio ei ole paras kaikilla p:n arvoilla. Kuitenkin paras konfiguraatio tietyllä p:n arvolla menestyy hyvin myös kaikilla muilla p:n arvoilla.

Vertailuna edellisestä kaaviosta todettakoon, että p:n arvoilla 0,05:stä 0,08:aan paras konfiguraatio poikkeaa vain 2,4% p=0,03 parhaasta ja 1,7% p=0,04:n parhaasta. Vastaavasti p=0,03 ja p=0,04:n paras konfiguraatio {1, 4, 8, 9, 10} on kuormalla 0,05 3. paras, 0,06:lla 4. paras, 0,07:lla 6. paras ja 0,08 10. paras.

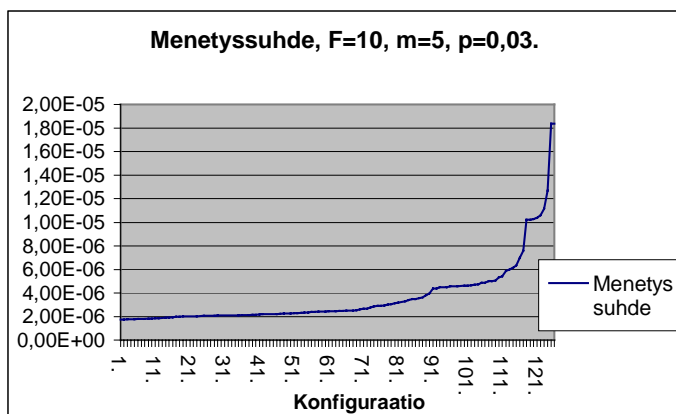
Loppupäässä on samalla tavalla pieniä eroja, vaikka huonot erottuvat omassa luokassaan. FIFO-puskurin tapaan mallinnettavat konfiguraatiot {1, 2, 3, 4, 5} ja {1, 3, 5, 7, 9} ovat huonoimpia kaikissa tapauksissa. Kaikissa tapauksissa, joissa konfiguraatio on tyyppiä {1, 2, 3, 4, m₅}, alkupään lyhyet viivelinjat ja niiden jaksollisuus ovat hallitsevia ja määräävät konfiguraatioille huonon suorituskyvyn.

Kun F, m ja p on kiinnitetty, analyttisesti laskettuja menetyssuhteita vertaamalla voidaan todeta seuraavaa.

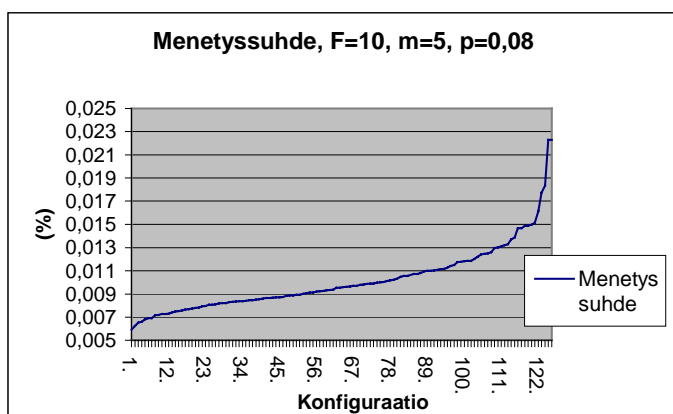
Kohdan 2.3 tapauksen 1 mukaiset konfiguraatiot ovat huonoudessaan omaa luokkaansa. Muiden konfiguraatioiden suorituskyvyllä ei ole suoranaista yhteistä sääntöä.

Kuvissa 7 ja 8 on vertailtu menetyssuhteen muodostamaa kuvaajaa erilaisilla kuorman arvoilla. Voidaan huomata, että mitä pienempi kuorma, sitä enemmän on havaittavissa selkeitä ryhmittymiä eri konfiguraatioryhmien paremmuudelle.

Ylemmässä kuvassa suuri hyppy menetyssuhteessa tapahtuu 118. konfiguraation kohdalla. Tästä lähtien kaikki konfiguraatiot ovat "lähes" jaksollisia eli eroavat huonoimmista konfiguraatioista, {1, 2, 3, 4, 5} ja {1, 3, 5, 7, 9} vain viimeisen viivelinjan osalta. Alemmassa kuvassa suuremmalla kuormalla taas vastaava ei toteudu niin selkeästi.



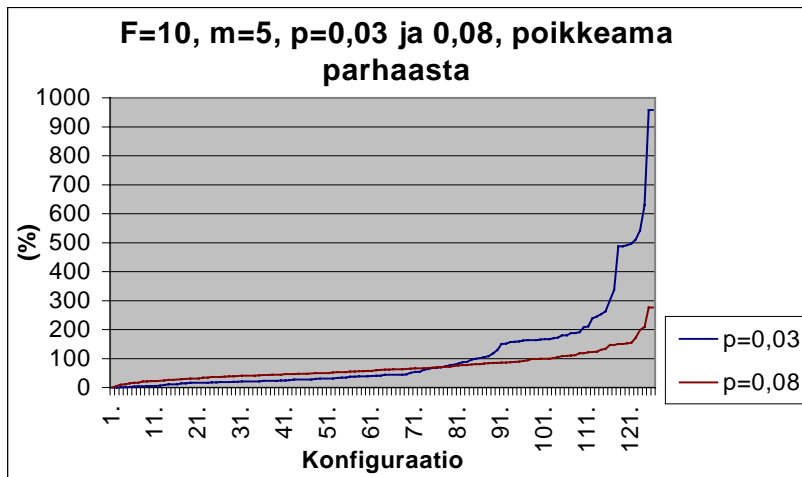
Kuva 7.



Kuva 8.

Erot menetyssuhteille kehittyvätkin eri tavalla eri kuormilla. Pienellä kuormalla parhaat ovat todella lähellä toisiaan, mutta huonoihin konfiguraatioihin on suuret hyppäykset. Toisaalta, mitä suuremmaksi kuorma kasvaa, sitä enemmän konfiguraatioilla on eroja parhaassa päässä, kun taas huonoimmassa päässä erot ovat pienempiä kuin pienemmällä kuormalla.

Kuvassa 9 on selvitetty edellistä havaintoa. Parametrina on konfiguraatioiden poikkeamat parhaasta konfiguraatiosta kyseisellä kuormalla. Huomataan, että 76. konfiguraation jälkeen suuremman kuorman erot parhaasta jäävät prosenteissa pienemmäksi verrattuna pienempään kuormaan.



Kuva 9. Menetyssuhteen muuttuminen eri konfiguraatioiden välillä

3.4 Hyvät ja huonot konfiguraatiot

Yhteenvedonä edellisestä todetaan, että selkeästi seuraavilla algoritmeilla valitut konfiguraatiot ovat yleensä hyviä.

- 1) Voidaan sanoa että mahdollisimman suurella F:n arvolla saadaan paras tulos. Kuitenkaan huonoin konfiguraatio syvyydellä F ei ole parempi kuin paras syvyydellä $<F$. Sama pätee m:n valintaan.
- 2) Hyvällä konfiguraatiolla on sekä pitkiä että lyhyitä viivelinjoja.
- 3) Kun valitaan viivelinjat 1,2, F-1 ja F sekä loput m-4 linjaa satunnaisesti, saavutetaan hyvä konfiguraatio.
- 4) Kokonaan satunnainen konfiguraatio tuottaa hyvän tuloksen, jos vain karsitaan säännölliset ja symmetriset, FIFO-puskurin tapaan mallinnettavat tapaukset pois.
- 5) Jos laitetaan viivelinjat alku- ja loppupäähän, saavutetaan kohtalainen konfiguraatio, niin kuitenkin, että on enemmän pitkiä kuin lyhyitä viivelinjoja.

Vastaavasti huonoiksi osoittautuu selkeästi

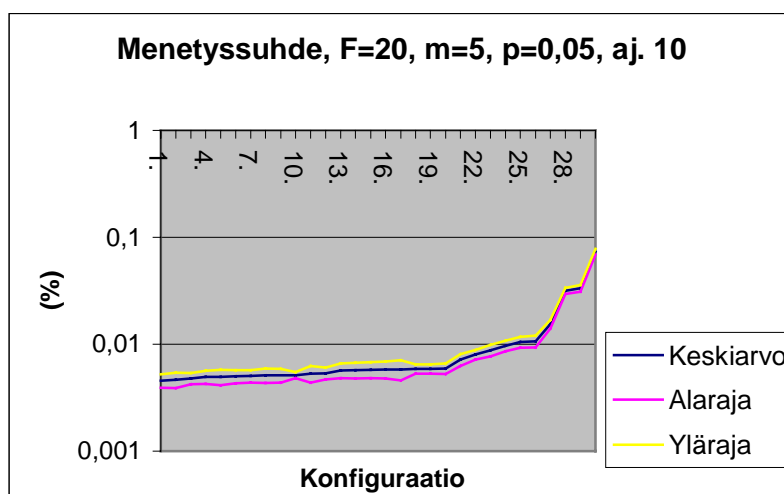
- 1) Jaksollista tyyppiä, esim. {1, 2, 3, 4, 5} ja {1, 3, 5, 7, 9}, ovat konfiguraatiot.
- 2) Jaksollisia konfiguraatioita lähellä olevat tapaukset, etenkin ne joilla vain viimeinen viivelinja poikkeaa jaksollisesta muodostelmasta.
- 3) Mitä lyhempi on viimeinen viivelinja, sitä todennäköisemmin konfiguraatio on huono.

4. Menetyssuhteen analysointi simuloiden

Koska matemaattisen ratkaisun laskemiseen kuluva aika Mathematicalla kasvaa melko suureksi F:n kasvaessa yli 10:nen, suurempien F:n arvojen tutkimiseen on käytetty simulointia. Samoin, koska erilaisten konfiguraatioiden määrä kasvaa, ei olla tutkittu kunkin kiinteän F:n ja m:n arvolla muodostettuja kaikkia mahdollisia konfiguraatioita, vaan on valittu tiettyjä. Simulointi on suoritettu Jaakko Tiaisen erikoistyössään tekemällä simulaattorilla.

Valitaan F:n arvoksi 20 ja m:ksi 5, 6 ja 7. Valitaan konfiguraatioita, jotka ovat satunnaisia, jaksollisia tai kohdassa 3 todettujen algoritmien mukaisia. Lisäksi kokeillaan algoritmia, jossa viivelinjat muodostetaan 1:stä ylöspäin ja F:sta alaspäin lisäämällä ja vähentämällä alkulukuja.

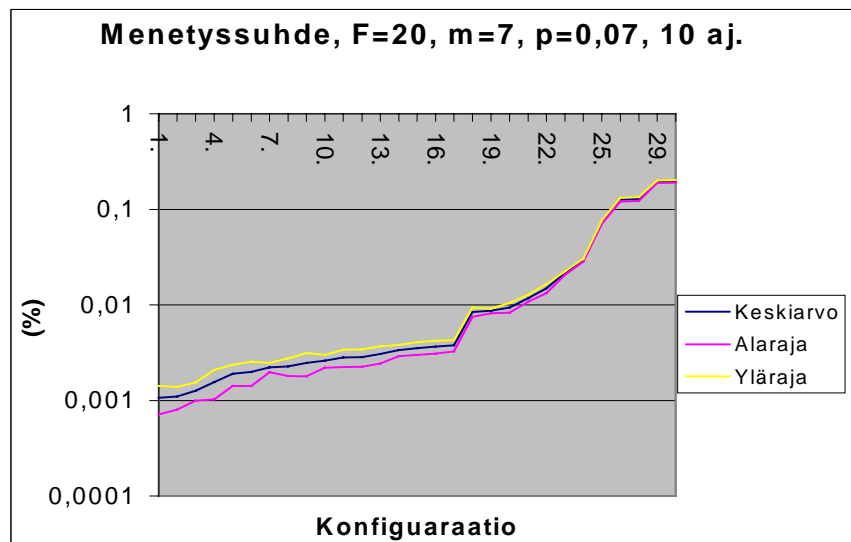
Simulointiajaksi on kiinnitetty 1 000 000 aikayksikköä. Simuloinnit on tehty 10:lla ajokerralla, mikä antaa tuloksille jonkin verran satunnaisuutta. Jos simulaattoriin tulee liian vähän paketteja aikayksikössä suhteessa puskurin kapasiteettiin, simulaattorin tarkkuus häviää koska menetyssuhteeksi saadaan usein nolla. Tämän takia eri m:n arvoilla on käytetty eri p:n arvoa.



Kuva 10. Menetyssuhteen kehittyminen 95% luottamusvälillä, konfiguraatiot liitteessä 3

Tapauksessa F=20, m=5, 3. kappaleen algoritmeista parhaiten menestyy tapaukset 3 ja 4, eli kun viivelinjat valitaan satunnaisesti, tai siten, että mukana on pituudet 1, 2, F-1 ja F. Huonoimpia ovat taas jaksolliset. Vaikka 28. konfiguraatio on satunnaisesti valittu, se kuitenkin sattumalta muistuttaa viimeistä viivelinjaa lukuunottamatta jaksollista konfiguraatiota. Tästä johtuu sen huono menetyssuhde. Parhaiten konfiguraatioiden erot tulevat paremmin esille logaritmisella asteikolla. Oheisesta kuvasta 10. ne voi kuitenkin havaita melko pieniksi.

Vastaavasti on tutkittu kuudella ja seitsemällä viivelinjalla menetyssuhdetta. Mukaan on lisäksi otettu tapauksia, joissa juuri viimeinen viivelinja poikkeaa jaksollisesta. Kuorma p on muutettu arvoon 0,06 kuudella viivelinjalla ja arvoon 0,07 seitsemällä viivelinjalla. Muutosta kuvattu logaritmisella asteikolla. Taulukot ovat liitteessä 4 ja tuloksien kuvaaja kuvassa 11.



Kuva 11. Menetyssuhde paremmuusjärjestyksessä 95% luottamusvälillä. Vastaavat konfiguraatiot liitteessä 4.

Tässäkin jaksolliset ja “melkein” jaksolliset konfiguraatiot erottuvat omina kategorioinaan. Muut konfiguraatiot muodostuvat samaan tapaan kuin tapauksessa $F=20, m=5$. Satunnaisuus parhaimpien kesken on ilmeistä. Yllättäen aikaisemmin kehitelty metodi konfiguraation muodostamisesta alkulukuja lisäämällä osoittautuu nyt melko hyväksi valinnaksi. Sen voi kuitenkin laskea samaan luokkaan satunnaisten konfiguraatioiden kanssa.

Muutos parhaassa päässä taas on suurempaa isommalla kuin pienemmällä kuormalla, samoin kuin analyttisissä tuloksissa. Kuitenkin isollakin kuormalla erottuu selviä ryhmiä, kuten suuri harppaus 18. konfiguraatiosta lähtien.

Kappaleessa 3 valitut algoritmit tuntuvat siis ainakin näissä tapauksissa oikeilta. Huonoihin tulisi lisätä tapaus “melkein” jaksollinen tai sarjallinen.

5. Laskeminen Mathematicalla eri parametrein

Menetyssuhteen laskemiseen kuluvaa aikaa on mitattu eri parametrein. Muunneltavia parametrejahan ohjelmassa on viivelinjojen lukumäärä m ja niiden konfiguraatio, sisääntulolinjojen lukumäärä n ja kuorma p .

Ohjelmaa kokeilemalla huomataan, että sisääntulolinjojen lukumäärä ja kuorma ei vaikuta laskuaikaan. Tämä johtuu siitä että nämä parametrit ovat laskussa vain yksittäisiä arvoja, eivätkä ne vaikuta yhtälöiden määrään tai lausekkeiden pituuteen.

F :n suuruus taas vaikuttaa suoraan generoitavien yhtälöiden määrään. Tiloja luodaan aluksi 2^F kappaletta. Laskuaikaan vaikuttaa siis olennaisesti F :n suuruus, mutta toisaalta viivelinjojen määrä ja konfiguraatio.

Konfiguraatio	Aika	Konfiguraatio	Aika
1 10	3 min 39 s	1 2 13	81 min,34 s
1 2 10	1 min 36 s	1 2 11 13	28 min, 1s
1 2 7 9 10	33s	1 2 5 6 7 9 10 13	3 min, 42 s
1 2 3 4 5 6 7 8 9 10	47s	1 2 3 4 5 6 7 8 9 10 11 12 13	9 min, 39 s
1 11	11 min 28 s	1 2 14	>5 h
1 2 11	5 min, 29 s	1 2 4 7 8 11 13 14	8 min 44 s
1 2 7 11	49 s	1 2 3 4 5 6 7 8 9 10 11 12 13 14	23 min
1 2 7 8 10 11	55 s	1 2 15	>6h
1 2 4 8 10 11	55 s	1 2 4 5 7 10 11 14 15	20 min 51s
1 2 3 4 5 6 7 8 9 10 11	1 min 36 s	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	54 min 15 s
1 2 12	21 min 34 s		
1 2 7 12	1 min 33 s		
1 2 3 4 5 6 7 8 9 10 11 12	3 min 50 s		

Taulukko 4. Laskenta-aika eri konfiguraatiolle, $p=0,05$ ja $N=10$

Ohjelma etsii oikeat tilasiirtymät ja luo niistä tilasiirtymämatriisin. Kun puskurin syvyys kasvaa, tähän matriisiin tulee yhä enemmän tyhjiä paikkoja ja listoja. Tämä hidastaa laskemista oleellisesti.

Yllättäen laskuaika ei kasva tasaisesti kun viivelinjoja lisätään kiinteällä F :llä. Pisin laskuaika on konfiguraatiolla $\{1,F\}$. Konfiguraatiolla, jossa on mukana kaikki viivelinjat 1:stä F :ään, on keskinopea laskuaika. Nopein aika taas tulee jonnekin näiden tapausten välille, suunnilleen sinne missä viivelinjoja on $F/2$ kappaletta. Kohtuulliset laskenta-ajat saavutetaan siis kun F on alle 15.

6. Simulaattorin suorituskyvystä

Simulaattori on toteutettu C++:lla . Tarkastellaan simulaattorin laskentanopeutta ja kilpailukykyä analyttiseen laskuriin verrattuna.

Simulaattorissa voi parametreina muuttaa viivelinjojen määrää, niiden konfiguraatiota, sisään tulolinjojen määrää ja kuormaa, yhtä lailla kuin analyttisessäkin laskurissa. Lisäksi voidaan muuttaa ajojen lukumäärää ja simulointiaikaa. Kaikissa ajoissa on käytetty miljoonaa aikayksikköä simulointiaikana.

Simulointeja ajamalla voidaan todeta, että simulointiaika ei riipu viivelinjojen määrästä eikä pisimmän viivelinjan pituudesta, kuten analyttisellä menetelmällä laskiessa. Odotetusti simulointiaika kasvaa lineaarisesti ajoja lisättäessä.

Parametri, joka muuttaa jonkin verran simulointiaikaa, kuorma. Oheisessa taulukossa huomaa pienen hidastumisen simulointiajassa, kun kuormaa kasvatetaan runsaasti. Tämä johtuu siitä, että suuremmalla kuormalla on enemmän käsiteltäviä tapahtumia.

	Ajoja	Aika (s)
p=0,05	10	10
	100	100
	500	520
p=0,07	10	12
	100	130
	500	605
p=0,10	10	13
	100	142
	500	670
p=0,20	10	15
	100	160
	500	823

Taulukko 5. Konfiguraatio 1 2 7 9 10

Tällä hidastumisella ei ole kuitenkaan varsinaista merkitystä, koska simuloinnit on suoritettu yleensä pienemmillä kuormilla, 0,01:stä 0,10:aan. Tällä välillä konfiguraation suorituskyvyissä on eroja, kun taas suurella kuormalla ei.

Toisaalta, simulaattorin suorituskykyä arvioitaessa tärkeä tekijä on tuloksen tarkkuus. Jos eroja parhaiden konfiguraatioiden välillä haluaa tehdä, tulee simulaatioita suorittaa satoja.

	Ajoja	Aika (s)	Keskiarvo	Poikkeama lasketusta (%)
p=0,05	10	10	0,00574	5,67
	100	100	0,00614	0,947
	500	520	0,00613	0,792
p=0,07	10	12	0,13315	0,594
	100	130	0,13367	0,212
	500	605	0,13440	0,336

Taulukko 6. Konfiguraatio 1 2 7 9 10

Seuraavassa taulukossa on vertailtu poikkeamia parhaista konfiguraatioista. Voidaan siis havaita, että pienellä kuormalla välttämättä 500 ajoakaan ei anna luotettavaa tietoa paremmuusjärjestyksestä. Mitä suuremmaksi kuorma kasvaa, sitä suuremmat ovat erot parhaimpien kesken ja vähemmällä ajoilla voidaan järjestys laatia.

	0,03	0,04	0,05	0,06	0,07	0,08
1.	0	0	0	0	0	0
2.	0,0407	0,0873	2,62	10,2	10,3	5,58
3.	0,552	1,76	2,91	11,7	16,5	9,93
4.	1,01	3,08	8,49	12,5	20,9	11,2
5.	2,43	4,27	12,1	18,6	20,9	14,5
6.	3,38	5,83	12,9	24,0	21,8	16,4

Taulukko 7. Kuuden parhaan konfiguraation menetyssuhteiden poikkeamat prosentteina kaikista parhaimmasta konfiguraatiosta kuormilla 0,03:sta 0,08:aan.

Simulaattorilla on rajoituksia pieniä menetyssuhteita simuloitaessa. Kun kuorma on pieni ja viivelinjojen määrä suuri, esimerkiksi p=0,05 ja m=20, menetyssuhde jää analyttisesti laskemalla hyvin pieneksi. Simulaattori taas ei anna kunnollista tulosta ollenkaan, koska pienin havaittu simuloinnin antama menetyssuhde on kokoluokkaa 10^{-4} . Tästä pienemmät ovat vain nolliä.

Simulaation ja analyttisen laskemisen vertailussa voidaan todeta, että tarkempia ja nopeampia tuloksia saadaan analyysistä kun puskurin syvyys on 11 tai vähemmän. Muilla parametreilla ei ole oleellista merkitystä. 11:sta eteenpäin analyttisen laskurin laskuaika kasvaa suuresti, jolloin simulointi on ainoa vaihtoehto. Tästä eteenpäin tulee joko tyytyä epätarkempiin tuloksiin simuloimalla tai käyttää runsaasti aikaa laskentaan.

7. Yhteenveto

Tutkinnassa ei löytynyt yksiselitteisiä vastauksia optimaalisen viivelinjakonfiguraation saavuttamiseksi. Kuitenkin pystytään sanomaan tarkasti, minkälaiset konfiguraatiot ovat todella huonoja. Kun nämä konfiguraatiot karsitaan ja valitaan edellä esitetyillä algoritmeilla viivelinjat, voidaan ainakin melko hyvä konfiguraatio saavuttaa.

Toisaalta, parhaiden konfiguraatioiden erot toisiinsa nähden ovat usein aika pieniä ja näiden erojen tuottamat lopulliset edut ovat luultavimmin myös vähäpätöisiä kokonaissysteemissä. Näitä pieniä eroja saa paremmin selville analyttisellä laskennalla, mutta suurilla F :n arvoilla se vaatii ohjelman tehostamista esimerkiksi tilamatriisia tiivistämällä.

Edellä on tarkasteltu suorituskyyä lähinnä vain menetyssuhteen kannalta. Suorituskyyanalyyysissa tarkasteltavaksi parametriksi voisi ottaa mukaan pakettien kokeman viiveen.

Todellisuudessa viivelinjojen pituus ja määrä on rajallinen käytännön toteutuksien takia. Kuitenkin on hyvin mahdollista toteuttaa puskurointi siten, että viivelinjoja on enemmän ja ne ovat pitempiä, kuin tässä työssä on tutkittu. Kuitenkin työssä todettuja algoritmeja voidaan soveltaa myös muun kokoisille kokonaisuuksille.

Lähdeluettelo

- [1] Cadro, P., Gravey A., Guillemot, C., Performance evaluation of an optical transparent packet switch, France Telecom CNET, 1996
- [2] Tiainen, Jaakko, Teletekniikan erikoistyö Optisen pakettimultiplekserin kapasiteetin simulointi, 2000
- [3] Hyytiä, Esa, Optical delay lines in a packet multiplexer, sisäinen muistio

Liitteet

Liite1. Mathematica-paketti menetyssuhteen laskemiselle

```
BeginPackage["PLossDelayLines`"]
```

```
PLossDLs::usage = "PLossDLs[F, l, n, p]  
evaluates a packet loss probability for optical packet switch that solves  
buffering using delay lines. Maximum depth of puffer is a parameter F and  
konfiguration of delay lines is a list l. There are n incoming lines with  
payload p."
```

```
Begin["`Private`"]
```

```
PLossDLs[F_, l_, n_, p_] := Module[  
  {m, a, b, p0, kokoelma, oikeet, oktaulu, yhtälölk, saraklkm, isoin, onevec,  
  bvec, amat, loppu, ea, ptn, pivec, parittomat, indeksit, tnt, es, kysely,  
  sijoittelu, joo, putsaus, pt},
```

```
  m = Length[l];
```

```
  (*Luodaan kaikki tilat sisältävä lista*)
```

```
  a = Table[x, {x, 0, (2^F)-1}];
```

```
  (*parillisella luvulla oikeassa reunassa 1, joka voidaan palvella*)
```

```
  b = (a /. (x_ /; OddQ[x]) -> x - 1);
```

```
  (*aikasiirto eli liikutetaan 1-0-jonoa oikealle eli jaetaan luvut kahtia*)
```

```
  p0 = b/2;
```

```
  (*tutkitaan onko luvun a binääriesityksen l:s luku 1 vai 0*)
```

```
  kysely[lp_, a_] := Module[{z},
```

```
    If[Mod[Nest[Function[x, Quotient[x, 2]], a, lp - 1], 2] == 0, 0, 1];
```

```
  (*sijoitellaan vapaisiin paikkoihin paketteja,
```

```
  tehdään eri taulukot kun sijoitetaan 1,2 ..m pakettia*)
```

```
  sijoittelu[a_] := Module[{z}, z = a; For[i = m, i > 0,
```

```
    If[kysely[l[[i]], a] == 0,
```

```
      z = a + 2^(l[[i] - 1)]; i--]; z];
```

```
  joo[q_] := NestList[sijoittelu, q, m];
```

```
  (*taulukko jossa 1. sarakkeessa lähtötilanne,
```

```
  2. sarakkeessa tila johon päädytään tn p0,
```

```
  3.sarakkeessa tila johon päädytään tn p1 jne*)
```



```

kokoelma = Transpose[Map[joo, p0]];
kokoelma = Transpose[Prepend[kokoelma, a]];

(*käydään läpi tiloja rekursiivisesti, niin kauan ettei lista toteutuvista tiloista
muutu*)
putsaus[x_] := Union[Flatten[Append[x, Table[kokoelma[[x[[i]] + 1]],
{i,Length[x]}]]]];
oikeet= FixedPoint[putsaus, kokoelma[[1]]];

(*oktaulu jossa 1. sarakkeessa alkutila, sitten palveltu tilanne eli p0,
sitten p1,p2 ja p3..pN*)
oktaulu= Table[Part[kokoelma, oikeet[[i]] + 1], {i, Length[oikeet]};

yhtälölk = Length[oktaulu];
sarakl = Length[oktaulu[[1]]];
isoin = Max[oktaulu];

For[i = 1, i <= yhtälölk, i++,
  oktaulu = ReplaceAll[oktaulu, oktaulu[[i, 1]] -> t[i + isoin]];
For[i = 2, i <= sarakl, i++,
  oktaulu[[All, i]] = oktaulu[[All, i]]*pt[i - 2]];
loppu = Transpose[ Table[Table[Cases[oktaulu[[j]], pt[x_]*t[i]],
  {i, isoin + 1, isoin + 1 + yhtälölk}],
  {j, yhtälölk}]];
For[i = 1, i <= yhtälölk, i++, loppu[[i, All]] = loppu[[i, All]]/t[i + isoin]];

(*koska yhdessä aikavälissä voidaan sijoittaa puskuriin max m pakettia,
tulee pm.n sisältää myös loput tn.t aina siihen asti että paketteja saapuu N
kappaletta yhdessä aikayksikössä, ylimääräiset vain menetetään*)
loppu = ReplaceAll[loppu, pt[m] -> ptn];

(*lopullinen tilamatriisi*)
loppu = Table[Table[Apply[Plus, loppu[[i, j]], {j, yhtälölk}],
{i,yhtälölk}];

(*odotusarvo saapumisten lukumäärälle*)
ea = n*p;

(*saapumisprosessi binomijakautunut*)
pt[i_] := Binomial[n, i] p^i (1 - p)^(n - i);

ptn = 1 - Sum[pt[i], {i, 0, m - 1}];

onevec = Table[1, {yhtälölk}];
bvec = Table[0, {yhtälölk - 1}];

(*lisätään tilayhtälöihin normeerausyhtälö*)
bvec = Append[bvec, 1];
amat = Take[loppu - IdentityMatrix[yhtälölk], yhtälölk - 1];

```

```

amat = Append[amat, onevec];

(*ratkaistaan tilayhtälöt*)
pivec = LinearSolve[amat, bvec];
parittomat = Select[oikeet, OddQ];
indeksit = Flatten[Table[Position[oikeet, parittomat[[i]]],
{i,Length[parittomat]}]];
tnt = Table[pivec[[indeksit[[i]]]], {i, Length[indeksit]}];

(* odotusarvo palveluille paketeille aikayksikössä eli niiden tilojen summa
joista paketti pääsee palveluun*)
es = Apply[Plus, tnt];
(ea - es)/ea
]

End[]
EndPackage[]

```

Liite 2. $F=10$, $m=5$, $p=0,05$, lasketut menetyssuhteet

	Konfiguraatio	Menetyssuhde		Konfiguraatio	Menetyssuhde
1.	1 2 7 9 10	6,082E-05	64.	1 6 7 8 10	0,000125752
2.	1 3 8 9 10	6,241E-05	65.	1 6 7 8 9	0,000125752
3.	1 4 8 9 10	6,259E-05	66.	1 3 4 6 10	0,000127818
4.	1 2 6 9 10	6,598E-05	67.	1 2 5 6 7	0,000129519
5.	1 2 6 8 9	6,820E-05	68.	1 4 5 6 9	0,000130091
6.	1 6 7 9 10	6,868E-05	69.	1 3 4 7 10	0,000130735
7.	1 6 8 9 10	7,054E-05	70.	1 4 6 8 10	0,000132003
8.	1 2 7 8 10	7,478E-05	71.	1 2 4 7 10	0,000134802
9.	1 3 6 7 10	7,531E-05	72.	1 3 5 8 9	0,00013536
10.	1 3 6 7 9	7,600E-05	73.	1 4 7 8 9	0,00014034
11.	1 4 6 9 10	7,613E-05	74.	1 2 4 5 10	0,000140809
12.	1 5 8 9 10	7,656E-05	75.	1 3 4 6 9	0,0001414
13.	1 3 7 8 10	7,708E-05	76.	1 4 7 9 10	0,000142188
14.	1 5 7 8 10	0,0000794	77.	1 4 7 8 10	0,000144352
15.	1 5 6 8 9	8,122E-05	78.	1 2 3 8 10	0,000144781
16.	1 2 5 9 10	8,188E-05	79.	1 5 6 7 8	0,000145934
17.	1 2 6 7 9	0,0000822	80.	1 4 5 6 8	0,000148372
18.	1 3 6 8 9	8,298E-05	81.	1 3 5 8 10	0,000148928
19.	1 5 6 8 10	8,318E-05	82.	1 2 4 5 9	0,000149732
20.	1 2 5 7 10	8,394E-05	83.	1 2 3 9 10	0,000152456
21.	1 4 5 9 10	0,0000841	84.	1 2 3 7 10	0,000152864
22.	1 2 6 7 10	8,427E-05	85.	1 5 7 9 10	0,000153541
23.	1 3 7 8 9	8,445E-05	86.	1 3 5 6 10	0,000153558
24.	1 4 5 8 10	8,481E-05	87.	1 2 4 6 9	0,00015382
25.	1 2 6 8 10	0,0004819	88.	1 2 3 8 9	0,000160636
26.	1 5 6 9 10	8,483E-05	89.	1 3 5 6 9	0,00016182
27.	1 3 4 9 10	8,486E-05	90.	1 2 3 7 9	0,000162409
28.	1 2 4 9 10	8,495E-05	91.	1 3 4 6 7	0,000165085
29.	1 2 5 7 8	8,591E-05	92.	1 2 4 6 7	0,000168455
30.	1 3 6 7 8	8,596E-05	93.	1 2 4 5 8	0,000169606
31.	1 3 6 9 10	8,792E-05	94.	1 2 4 6 10	0,000170354
32.	1 5 7 8 9	8,845E-05	95.	1 3 4 6 8	0,000172664
33.	1 3 4 8 10	8,870E-05	96.	1 3 4 5 10	0,000172944
34.	1 2 8 9 10	9,044E-05	97.	1 2 3 7 8	0,000177165
35.	1 4 6 7 10	9,053E-05	98.	1 3 5 6 8	0,000180417
36.	1 2 7 8 9	9,276E-05	99.	1 2 3 6 10	0,000184389
37.	1 2 5 7 9	9,356E-05	100.	1 3 4 5 9	0,000184877
38.	1 4 6 8 9	9,443E-05	101.	1 2 4 5 7	0,000190511
39.	1 2 4 8 9	9,534E-05	102.	1 2 3 6 8	0,000191955
40.	1 3 6 8 10	9,545E-05	103.	1 4 5 6 7	0,000199494
41.	1 3 4 8 9	9,604E-05	104.	1 3 5 9 10	0,000201474
42.	1 4 6 7 9	9,616E-05	105.	1 3 4 5 8	0,000204735
43.	1 4 5 8 9	0,0009865	106.	1 3 5 6 7	0,000209381
44.	1 4 5 7 10	9,958E-05	107.	1 2 3 6 7	0,00021383
45.	1 2 6 7 8	0,0000687	108.	1 2 3 6 9	0,000218775
46.	1 2 5 8 10	0,0001015	109.	1 2 3 5 10	0,000224792
47.	1 5 6 7 10	0,0001019	110.	1 2 4 6 8	0,000237364

48.	1 4 6 7 8	0,0001058	111.	1 2 3 5 9	0,000242053
49.	1 2 5 8 9	0,0001059	112.	1 2 4 5 6	0,000243389
50.	1 2 5 6 8	0,0001067	113.	1 3 4 5 7	0,000244096
51.	1 2 5 6 10	0,0001069	114.	1 2 3 5 8	0,000256621
52.	1 2 4 8 10	0,0001072	115.	1 2 3 5 6	0,000316014
53.	1 3 4 7 9	0,0001074	116.	1 3 5 7 10	0,000328266
54.	1 2 5 6 9	0,0001102	117.	1 2 3 5 7	0,000334007
55.	1 4 5 7 8	0,0001113	118.	1 3 5 7 8	0,000334283
56.	1 5 6 7 9	0,0001116	119.	1 3 4 5 6	0,000342059
57.	1 2 4 7 8	0,0001119	120.	1 2 3 4 10	0,000348719
58.	1 4 5 7 9	0,0001123	121.	1 2 3 4 9	0,000362676
59.	1 2 4 7 9	0,0001164	122.	1 2 3 4 8	0,00038634
60.	1 3 4 7 8	0,0001181	123.	1 2 3 4 7	0,000428895
61.	1 7 8 9 10	0,0001181	124.	1 2 3 4 6	0,000514853
62.	1 4 5 6 10	0,0001202	125.	1 3 5 7 9	0,000736693
63.	1 3 7 9 10	0,0001228	126.	1 2 3 4 5	0,000736693

Liite 3. F=20, m=5, p=0,05, 10 ajon simuloinnit 95% luottamusvälillä

F=20, m=5, p=0,05		Konfiguraatio	Keskiarvo	Alaraja	Yläraja
1.	s	1 6 7 16 20	0,004576	0,003911	0,005241
2.	1,2+x+F-1,F	1 2 14 19 20	0,004660	0,003884	0,005437
3.	1,2+x+F-1,F	1 2 6 19 20	0,004799	0,004217	0,005381
4.	1,2+x+F-1,F	1 2 13 19 20	0,004956	0,004266	0,005645
5.	1,2+x+F-1,F	1 2 8 19 20	0,004958	0,004133	0,005783
6.	1,2+x+F-1,F	1 2 15 19 20	0,005020	0,004316	0,005725
7.	1,2+x+F-1,F	1 2 9 19 20	0,005058	0,004389	0,005727
8.	1,2+x+F-1,F	1 2 7 19 20	0,005137	0,004341	0,005934
9.	1,2+x+F-1,F	1 2 12 19 20	0,005140	0,004387	0,005893
10.	s	1 3 17 19 20	0,005154	0,004809	0,005498
11.	s	1 6 12 15 20	0,005319	0,004387	0,006251
12.	1,2+x+F-1,F	1 2 16 19 20	0,005361	0,004687	0,006034
13.	1,2+x+F-1,F	1 2 5 19 20	0,005703	0,004822	0,006583
14.	alkuluku	1 2 17 19 20	0,005740	0,004776	0,006704
15.	1,2+x+F-1,F	1 2 17 19 20	0,005799	0,004828	0,006771
16.	s	1 3 11 17 20	0,005835	0,004806	0,006864
17.	s	1 7 11 14 20	0,005838	0,004589	0,007087
18.	alkuluku	1 14 17 19 20	0,005894	0,005314	0,006474
19.	jakso	1 2 10 14 20	0,005902	0,005323	0,006480
20.	1,2+x+F-1,F	1 2 11 19 20	0,005957	0,005296	0,006619
21.	alkuluku	1 2 4 19 20	0,007163	0,006235	0,008091
22.	1,2+x+F-1,F	1 2 10 19 20	0,008016	0,007169	0,008864
23.	alku+loppu	1 2 18 19 20	0,008782	0,007670	0,009894
24.	alkuluku	1 2 4 7 20	0,009712	0,008580	0,010843
25.	jakso	1 8 12 16 20	0,010549	0,009293	0,011805
26.	alku+loppu	1 17 18 19 20	0,010696	0,009340	0,012051
27.	alku+loppu	1 2 3 19 20	0,015425	0,013966	0,016884
28.	s	1 5 9 13 20	0,031651	0,029535	0,033766
29.	jaksopaitsivika	1 4 7 10 17	0,033316	0,031401	0,035231
30.	alku+loppu	1 2 3 4 20	0,033351	0,030889	0,035814
31.	jaksopaitsivika	1 2 3 4 17	0,034516	0,032318	0,036714
32.	jakso	1 2 3 4 5	0,073472	0,068893	0,078050

Liite 4. $F=20$, $m=7$, $p=0,07$, simuloidut, 95% luottamusvälillä

		Konfiguraatio	Keskiarvo	Alaraja	Yläraja
1.	1,2,x,y,z,F-1,F	1 2 6 14 17 19 20	0,001071	0,000717	0,001424
2.	alkuluku	1 2 9 14 17 19 20	0,001100	0,000803	0,001397
3.	alkuluku	1 2 4 14 17 19 20	0,001271	0,000997	0,001545
4.	s	1 2 9 11 17 18 20	0,001557	0,001024	0,002090
5.	1,2,x,y,z,F-1,F	1 2 8 11 17 19 20	0,001898	0,001415	0,002382
6.	1,2,x,y,z,F-1,F	1 2 4 9 13 19 20	0,001987	0,001429	0,002546
7.	alkuluku	1 2 4 7 17 19 20	0,002229	0,001978	0,002479
8.	1,2,x,y,z,F-1,F	1 2 5 12 15 19 20	0,002285	0,001809	0,002760
9.	s	1 6 9 14 15 17 20	0,002472	0,001795	0,003149
10.	1,2,x,y,z,F-1,F	1 2 7 13 14 19 20	0,002614	0,002217	0,003012
11.	1,2,x,y,z,F-1,F	1 2 8 11 14 19 20	0,002813	0,002237	0,003389
12.	s	1 3 7 11 14 19 20	0,002844	0,002270	0,003417
13.	1,2,x,y,z,F-1,F	1 2 7 8 18 19 20	0,003072	0,002431	0,003713
14.	s	1 3 7 9 14 17 20	0,003372	0,002916	0,003829
15.	s	1 5 9 15 17 18 20	0,003544	0,002999	0,004089
16.	alkuluku	1 2 4 7 12 19 20	0,003658	0,003096	0,004221
17.	1,2,x,y,z,F-1,F	1 2 5 9 12 19 20	0,003782	0,003289	0,004275
18.	alku+loppu	1 2 3 17 18 19 20	0,008452	0,007487	0,009416
19.	s	1 4 5 6 11 13 20	0,008670	0,008208	0,009132
20.	alku+loppu	1 2 16 17 18 19 20	0,009389	0,008278	0,010500
21.	alku+loppu	1 2 3 4 18 19 20	0,011782	0,010783	0,012781
22.	jakso(s.y.t)	1 2 4 8 12 16 20	0,014924	0,013348	0,016499
23.	alku+loppu	1 15 16 17 18 19 20	0,021638	0,020746	0,022530
24.	alku+loppu	1 2 3 4 5 19 20	0,029485	0,028431	0,030538
25.	alku+loppu	1 2 3 4 5 6 20	0,073619	0,070576	0,076661
26.	jaksopaittivika	1 3 5 7 9 11 17	0,126170	0,120326	0,132015
27.	jaksopaittivika	1 2 3 4 5 6 9	0,128355	0,122348	0,134362
29.	jakso	1 3 5 7 9 11 13	0,195231	0,188565	0,201896
28.	jakso	1 2 3 4 5 6 7	0,196445	0,190586	0,202304