Loppuraportti

Linux Traffic Control-käyttöliittymä – Ryhmä paketti2

Helsinki 21.12.2004

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhmä

Fabian Fagerholm
Janne Johansson
Markku Manner
Niko Mikkilä

Asiakas

Jukka Manner

Johtoryhmä

Juha Taina
Marianne Korpela

Kotisivu

`http://www.cs.helsinki.fi/group/paketti2`

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.1	1.11.2004	Vedos

Sisältö

1 Johdanto

Paketti2-projekti kuuluu Helsingin yliopstion tietojenkäsittelytieteen laitoksen Ohjelmistotuotantoprojekti-kurssiin. Kyseessä on jatkoprojekti syksyn 2003 Paketti-projektille [Pr03]. Tavoitteena on tuottaa ohjelmisto Linux-ytimen kaistanhallinta-asetusten hallinnointiin graafisessa muodossa. Järjestelmä tulee Helsingin yliopstion tietojenkäsittelyteiteen laitoksen tutkimusryhmien käyttöön. Yliopisto julkaisee ohjelman GPL- [FSF91] tai LGPL-lisenssin [FSF99] ehdoin.

1.1 Projektin henkilöt

Projektiryhmään kuuluivat opiskelijat Fabian Fagerholm, Janne Johansson, Markku Manner ja Niko Mikkilä. Projektin ohjaajana toimi Marianne Korpela, kurssin vastuuhenkilönä Juha Taina ja asiakkaana Jukka Manner.

1.2 Projektin aikataulu

Projekti alkoi ryhmän ensimmäisestä tapaamisesta 6.9.2004. Projekti päättyi tiistaina 14.12.2004, jolloin projektiin liittyvä materiaali luovutettiin. Alkuperäisen suunnitelman loppumispäivämäärä oli perjantai 10.12.2004, joten projekti päättyi muutaman päivän alkuperäistä suunnitelmaa myöhemmin.

1.3 Dokumentin tarkoitus ja rakenne

Tässä dokumentissa projektiryhmä arvioi projektin kokonaisuuden, eri vaiheiden ja yhteistyön onnistumisia ja epäonnistumisia. Samalla mietititään mitä ryhmä olisi voinut tehdä eri tavalla tai tulisi tekemään eri tavalla seuraavassa projektissa.

Luvussa 2 käsitellään projektin onnistumista yleiseltä kannalta, työtuntien jakautumista ryhmän jäsenten kesken projektin eri vaiheissa ja projektiin vaikuttaneita tekijöitä. Luku 3 käsittelee eri vaiheiden onnistumisia tarkemmin ja luvussa 4 mitä projektissa olisi kannattanut tehdä toisin.

2 Projektin onnistuminen

Projektiryhmä pysyi suunnitellussa aikataulussa aina toiseksi viimeiselle viikolle asti. Tällöin tuli eteen suurempia ongelmia, jotka lopulta viivästyttivät projektin valmistumista 10. päivästä 14. päivään joulukuuta. Tarkempi analyysi ongelmien syistä ja aikataulun pitenemisestä, löytyy luvusta 4.

Projektissa käytettiin lineaarista prosessimallia, tosin projektin eri vaiheet limittyivät aikataulullisesti toistensa kanssa päällekkäin. Taulukossa 1 on esitetty miten aikataulu muut-

tui sunnitellusta aikataulusta. Tarkempi suunniteltu aikataulu löytyy Paketti2-ryhmän projektisuunnitelmasta [Pr04c].

Vaihe	Suunniteltu toteutusaika	Toteutunut aika
Määrittely	6.9.2004 - 6.10.2004	6.9.2004 - 6.10.2004
Suunnittelu	29.9.2004 - 27.10.2004	29.9.2004 - 27.10.2004
Toteutus	20.10.2004 - 1.12.2004	20.10.2004 - 13.12.2004
Testaus	27.10.2004 - 10.12.2004	27.10.2004 - 13.12.2004
Luovutus	24.11.2004 - 10.12.2004	24.11.2004 - 14.12.2004

Taulukko 1: Projektin suunniteltu ja toteutunut aikataulu.

Projektissa onnistuttiin toteuttamaan prioriteetin 1 asiat, kuten ne on ilmaistu määrittelydokumentissa [Pr04b], ja hyväksymistestaus saatiin toteutettua kuten se on määritelty projektiryhmän testaussuunnitelmassa [Pr04d]. Ylläpitodokumentista (englanniksi) saa tarkempaa tietoa ohjelman tilasta [Pr04a].

2.1 Työtuntijakauma

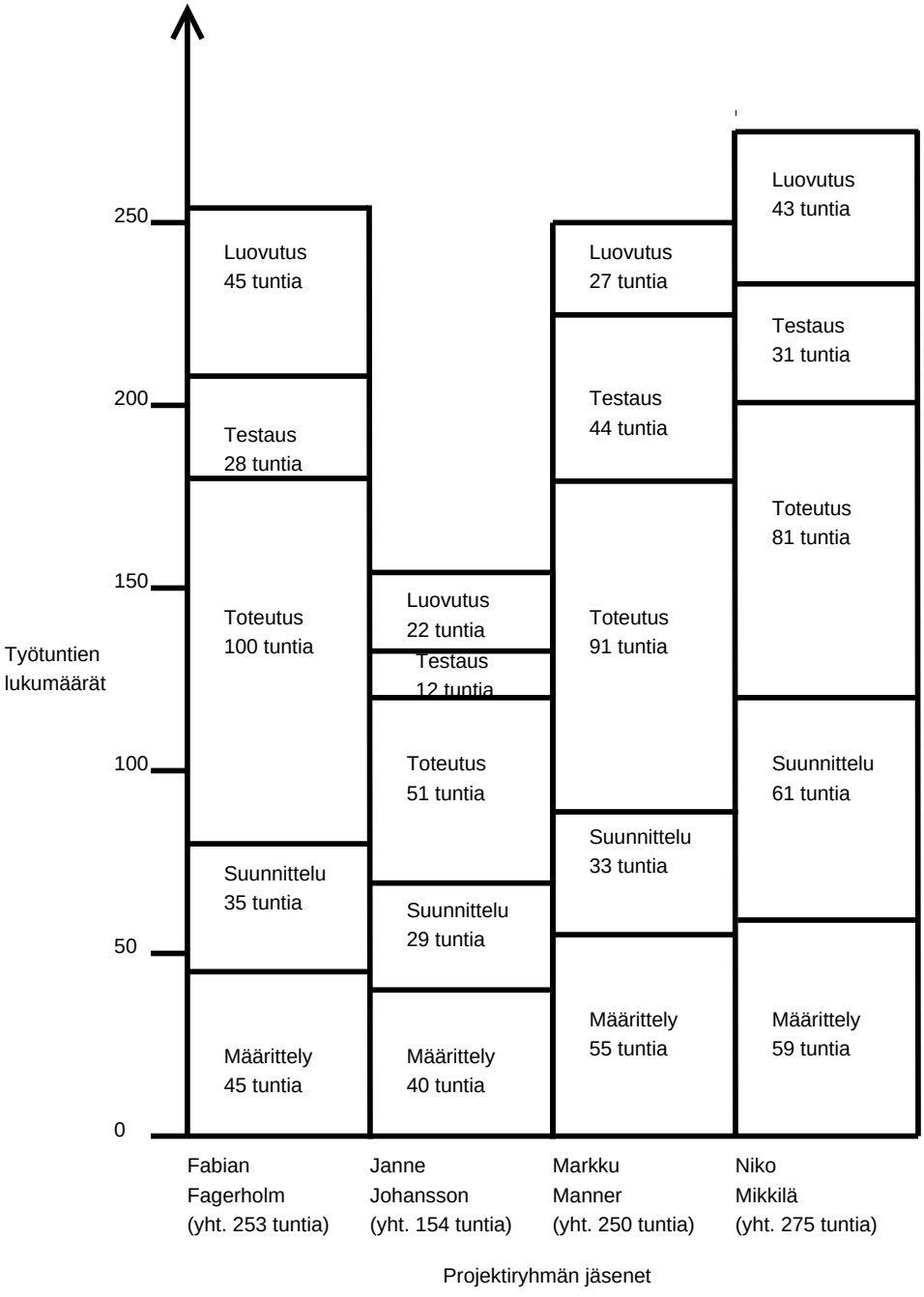
Kuvassa 1 on esitetty ryhmän tekemät tunnit jaoteltuna eri vaiheisiin. Vaiheet ja niiden päivämäärät ovat taulussa 1. Eri vaiheissa jäsenet tekivät myös muita töitä, eli esim. toteutuksen tunnit eivät pelkästään sisällä ohjelmointia vaan myös mm. jatkosuunnittelua, testausta ja dokumentointia.

Projektiryhmän jäsenten eri vaiheiden työtunnit ovat hyvässä suhteessa keskenään. Jäsenet käyttivät maksimissaan 10% enemmän omista työtunneistaan tietyssa vaiheessa verrattuna toisiin. Tämä ei ole erityisemmin suuri heitto. Projektiryhmän jäsenet tekivät muutenkin tasaisesti töitä suhteessa toisiinsa, kokonaistyötunnit ovat 10% sisään muiden työtunneista (Janne Johanssonin työtunnit eivät ole ajantasalla. Tämä johtuu unohduksesta pitää työtunnit ajantasalla, hän teki yhtä paljon töitä kuin loput ryhmän jäsenet).

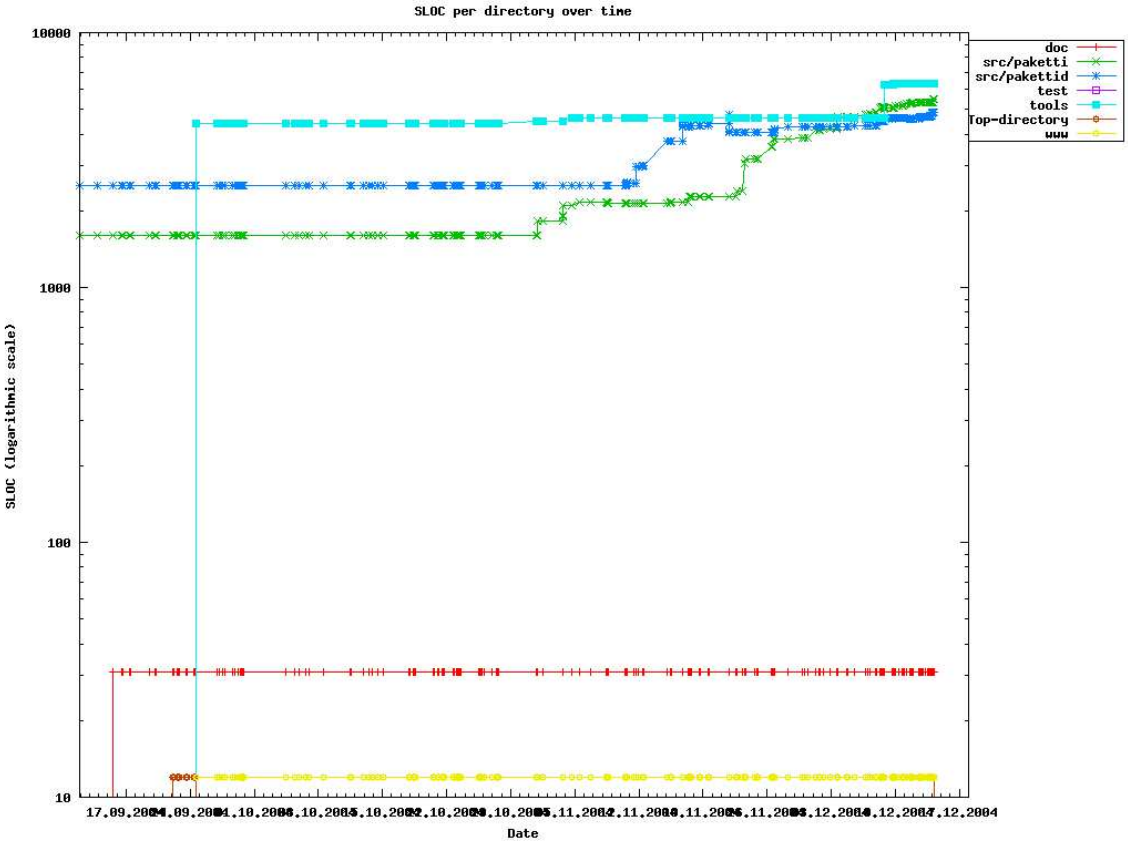
2.2 Ohjelmiston koko

Kuvassa 2 on esitetty projektin aikana tapahtunut SLOC-muutos. Vasemmassa laidassa on alkutilanne ja oikeassa laidassa lopulliset SLOC-arvot.

Arviomme SLOC-kokonaisuudeksi oli kohdallaan kontrollikomponentin osalta, mutta käyttöliittymän osalta petti suuremman kerran [Pr04c]. Alkuperäinen arviomme oli kontrollikomponentin osalta 4000 koodiriviä ja toteutunut oli 4091 koodiriviä. Käyttöliittymä oli arvioitu vievän 2800 koodiriviä, mutta toteutus kasvoi melkein kaksinkertaiseksi eli 5200 koodiriviin. Tämä ainakin osalta johtui Transform.java:n simppelistä toteutuksesta, joka oli nopeata tehdä, mutta koodirivejä tuli paljon (1214 koodiriviä). Koodirivejä saisi varmasti vähennettyä siistimällä koodia.



Kuva 1: Projektin jäsenten työtunnit eri vaiheissa



Kuva 2: Projektin aikaiset SLOC-arvot

2.3 Projektiin vaikuttaneita tekijöitä

Projektin sujumiseen positiivisesti vaikuttivat seuraavat asiat:

- Ryhmähenki oli moitteeton, projektin aikana ei tullut kertaakaan ongelmia projektiryhmän jäsenten kesken.
- Jäsenten välisessä yhteistyössä ei ilmennyt ongelmia.
- Töiden jaossa onnistuttiin hyvin, ryhmän jäsenet jaettiin henkilökohtaisten osaamisien mukaan oikeisiin tehtäviin.
- Projektin jäsenet panostivat projektiin koko projektin ajan ja hoitivat asiansa kunnolla.
- Jäsenet toimivat itsenäisesti annettujen tehtävien parissa, ja jokainen teki vähintään sitä mitä oli luvannut.
- Projektissa käytetyt työkalut osoittautuivat oikeiksi lopulta.
- Projektiryhmän tietotaso riitti projektiin, mikäli joku ei tiennyt jotain asiaa, muut pystyivät hyvin selittämään asian tai tiesivät opastaa mistä etsiä tietoa asiaan.

Negatiivisia asioita projektin sujumiseen on listattu seuraavaksi:

- Puutteellinen perehtyminen edellisen ryhmän työhön. Toteutusvaiheessa tuli eteen paljon odottamattomia ongelmia edellisen ryhmän kontrollikomponentin puolelta. Käyttöliittymän puolella ongelmat olivat vähäisiä.
- Projektissa lähdettiin liian nopeasti tekemään projektisuunitelmaa, määrittely- ja suunnitteludokumenttia. Aluksi olisi pitänyt perehtyä paljon syvällisemmin edellisen ryhmän toteutukseen sekä Linuxin Traffic Control-toteutukseen.
- Valittu prosessimalli ei välttämättä ollut sopiva projektiin. Iteratiivisella prosessimallilla ryhmä olisi voinut saada alussa paremman käsityksen edellisen ryhmän tuotoksesta ja näin määritellä ja suunnitella toteutuksen paremmin.
- Saatavilla oleva dokumentaatio oli hyvin suppea, ristiriitainen ja jopa virheellinen. Tämä koski edellisen ryhmän dokumentteja, mutta etenkin Linuxin dokumentaatiota.

3 Projektin eri vaiheiden onnistuminen

Projekti jaettiin vesiputousmallin mukaan viiteen eri vaiheeseen: määrittelyyn, suunnitteluu, toteutukseen, testaukseen ja luovutukseen. Seuraavaksi analysoidaan jokaisen vaiheen onnistumisia ja epäonnistumisia. Luovutusvaihetta ei kommentoida. Luovutus tapahtui tiistaina 14.12.2004, neljä päivää sunnitellusta myöhemmin.

3.1 Määrittely

Määrittely aloitettiin heti projekti alkaessa. Juteltuamme asiakkaan kanssa, saimme käsityksen mitä meidän pitäisi saada aikaan. Asiakkaalta saimme suhteellisen vapaat kädet suunnittelua varten. Määrittelyssä otimme huomioon asiakkaat toivomukset ja ryhmän jäsenten ehdotukset.

Määrittelyssä perehdyimme edellisen projektin saamaan ohjelmistoon päällisin puolin. Määrittelyssä tuli esille asioita, joita ryhmän jäsenet katsoivat tärkeiksi muuttaa, jotta ohjelmistosta saataisiin selkeämpi kokonaisuus ja helpommin ylläpidettävä. Määrittely onnistui hyvin, saimme kasaan järkevän kokonaisuuden toteutettavia ominaisuuksia.

Määrittelyssä olisi ollut parempi perehtyä edellisen ryhmän tuotokseen tarkemmin, jotta olisi ollut selkeämpi kokonaisuus lähteä määrittelemään toteutettavia asioita. Määrittelyssä olisi myös pitänyt perehtyä hyvin tarkasti Linuxin Traffic Control-ominaisuuksiin, jotta sen mahdollisuudet ja rajoitukset olisivat olleet selviä. Ryhmässä oli paljonkin kokemusta Linuxin Netfilter- ja iptables-toteutuksista (palomuuriasetukset). TC-ominaisuudet eivät olleet yhtä tuttuja, emmekä onnistuneet arvoimaan tarvitsemamme tietojen määrää, jota olisi pitänyt olla tämän vaiheen jälkeen.

3.2 Suunnittelu

Suunnittelu aloitettiin määrittelyn perusteella. Käyttöliittymän suunnittelussa ei ilmeentynyt ongelmia, edellisen ryhmän toteutus oli selkeä. Käyttöliittymään suunniteltiin suuri muutos muuttamalla XML DTD:stä laajennettuun XML Schemaan. Tämän muutoksen avulla virheellisten syötteiden tarkistaminen tuli helpommaksi kuten myös käyttöliittymän laajennettavuus.

Kontrollikomponentin suunnittelu jäi hieman vajaaksi lähinnä päällisen perehtymisen Pakettiryhmän kontrollikomponentin koodeihin. Suunnittelussa otettiin huomioon lukuisia tarkeitä osia, joiden myötä kontrollikomponentti muuttui huomattavasti.

Suunnitteluun olisi tarvinnut paljon tarkempaa tietoa ohjelmiston laadusta ja toiminnasta. Tällöin suunnittelu olisi ollut tarkempaa ja helpomaa. Muutenkin suunnittelumme jäi pinnalliselle tasolle. Olisi ollut tärkeätä suunnnitella paljon tarkemmin asiat, kooditasolle asti. Tällöin toteutus olisi ollut helpompaa, koska olisi ollut jo tarkempi ymmärrys sovelluksen toiminnasta, ja todennäköisesti oltaisiin saatu enemmän aikaiseksi. Jätimme kuitenkin asioita suunnittelematta riittävän tarkasti, koska uskoimme vielä tässä vaiheessa voivamme rakentaa olemassa olevan koodin päälle, samalla koodaustyylillä ja samoilla ratkaisuperiaatteilla.

3.3 Toteutus

Toteutusvaiheeseen lähdimme puutteellisen suunnittelun jälkeen. Tämä kostautui suurella määrällä yllättäviä ongelmia varsinkin kontrollikomponentin puolella. Toteutusvaiheen aikataulu petti ja kesti pidempään kuin oltiin suunniteltu.

Käyttöliittymän toteutuksessa ei ilmentynyt pahemmin ongelmia. Ainoastaan XML Schemaan siirtyminen kesti kauemmin kuin olettiin, mutta tämä ei vaikuttanut projektin aikatauluun. Käyttöliittymäpuolen toteutus oli varsin suoraviivaista, toisaalta tämä oli huomattavasti helpompi osa ohjelmistosta.

Kontrollipuolen toteutuksessa ilmentyi suuria ennalta arvaamattomia ongelmia. Jälkeen päin arviointuna kontrollikomponentti olisi voinut olla parempi aloittaa tyhjästä eikä jatkaa Paketti-ryhmän kontrollikomponentilla. Kyseinen kontrollikomponentti toimi hyvin Paketti-ryhmän käyttöliittymällä, mutta Paketti2-ryhmän määriteltyjen ominaisuuksien implementoiminen vanhaan kontrollikomponenttiin oli vaikeampaa kuin osattiin arvella. Tämä aiheutti projektin aikataulusta jälkeenjäämisen viimeisillä viikoilla.

Aivan toteutusvaiheen lopussa Linuxin Traffic Control-ominaisuuksien rajoitukset ja puutteellinen dokumentaatio haittasi työtä huomattavasti. Koska emme projektin alussa onnistuneet havaitsemaan nämä puutteet, aika ei enää riittänyt ongelmien selvittämiseksi ja oman dokumentaation tuottamiseksi esimerkkikoodin ja Linuxin oman lähdekoodin perusteella. Uskomme edellisen ryhmän törmänneen samaan ongelmaan, mutta heillä ei ollut tästä mainintaa dokumentaatiossa. Arviomme perustuu heidän koodissaan käytettyihin kompromissiratkaisuihin.

3.4 Testaus

Testausvaihe kesti pidempään kuin suunniteltuna, koska toteutusvaihe pidentyi. Testaus jäi tämän takia vajaaksi, tosin suurimmat ongelmat saatiin korjattua.

Testauksessa ei käyttöliittymän puolella ilmentynyt suurempia ongelmia. Pieniä ongelmia löytyi jatkuvasti, mutta nämä kaikki saatiin korjattua pienellä vaivalla. Myös kontrollipuolen testaus onnistui varsin mallikkaasti, ongelmia löytyi mutta ne saatiin korjattua.

Integraatiotestauksessa ilmentyi suurimmat ongelmat. Kontrollipuoli toteutettiin Paketti-ryhmän dokumentoinnin perusteella, joka kostautui sillä ohjelmiston toteutus ei täysin vastannut dokumentointia. Tämä jälleen olisi huomattu ajoissa, mikäli olisimme perehtynyt tarkemmin käyttöliittymän ja kontrollikomponentin koodeihin. Onnistuimme kuitenkin korjaamaan ison osan olemassa olevista matalan tason virheistä, sekä dokumentoituja että aiemmin tuntemattomia.

4 Projektin arviointi

Projektissa opimme tärkeyden tietää mitä on tekemässä ennen kuin lähtee tekemään sitä. Eli olisi pitänyt perehtyä jokaiseen koodiriviin Paketti-ryhmän toteutuksesta, jotta määrittely, suunnittelu ja varsinkin toteutus olisi mennyt sujuvammin. Ryhmän kokemattomuus jatkoprojekteista oli varmasti suuri vaikuttaja tähän.

Ryhmätyöskentely toimi moitteetta. Ryhmän jäsenet toteuttivat sovitut asiat ja kommunikointi toimi. Lopussa aikataulun pettäminen johtui pelkästään kunnollisen perehtymisen puuttumisesta, ei yksittäisen ryhmän jäsenen huonosta tai myöhästyneestä työstä.

Taulukossa 2 on esitetty uusi suuntaa antava aikataulu, jonka perusteella olisi voinut olla kannattavampaa lähteä tekemään projektia. Tässä olisi tarpeeksi aikaa perehtyä koodiin riittävästi ja sitten määritellä. Määrittelyn jälkeen suunniteltaisiin tarkemmin, kun tiedettäisiin tarkasti miten ohjelma toimii. Tämän jälkeen toteutusvaiheen ei tarvitsisi olla niin pitkä, sillä ongelmiin oltaisiin paremmin varauduttu.

Vaihe	Aloituspäivämäärä	Päättymispäivämäärä
Perehtyminen	6.9.2004	6.10.2004
Määrittely	6.10.2004	20.10.2004
Suunnittelu	13.10.2004	10.11.2004
Toteutus	10.11.2004	1.12.2004
Testaus	20.11.2004	10.12.2004
Luovutus	10.12.2004	10.12.2004

Taulukko 2: Uusi aikataulu.

Projektin arvioinnissa herää kuitenkin myös kysymys, oliko projektiryhmässä oikeasti riittävästi resursseja tämän projektin läpiviemiseksi. Edellisessä ryhmässä jäseniä oli yksi enemmän, ja käsittääksemme arvio oli, että ryhmämme todellakin voisi jatkaa siitä mihin edellinen ryhmä jäi. Valitettavasti edellinen ryhmä oli kuitenkin, nyt saatujen kokemusten perusteella, päässyt niin pitkälle kuin voi, ilman että koko tehtävään ottaisi toisenlaisen näkökulman.

Projektin aikana olemme löytäneet lukuisia yrityksiä tehdä vastaavanlaisia ohjelmistoja, ja jokainen yritys on kompastunut matkaan tai on toistaiseksi pysähtynyt samaan ongelmaan. Linuxin kaistanhallinta-asetusten hallintaan löytyy kaksi pääasiallista tapaa: komentorivipohjaisen tc-työkalun kautta, ja suoraan rtnetlink-rajapintaa käyttäen. Kumpikin tapa on hyvin suppeasti dokumentoitu. Tc-työkalu ei ole valmis, ja se tukee puutteellisesti ytimen tarjoamia mahdollisuuksia. Rtnetlink taas on erittäin monimutkainen ja vaatii syvällistä paneutumista käyttöjärjestelmäkoodiin.

Ratkaistava ongelma on mielestämme liian iso tehtävä yhdelle projektille. Ensin pitäisi ymmärtää ja ratkaista mahdolliset ongelmat Linux-ytimen TC-koodissa, jonka jälkeen kyseinen toiminnallisuus tulisi dokumentoida paremmin. Tämän jälkeen voisi kirjoittaa käyttäjätilan kirjasto (ks. [Ols02], [Mal99] ja [Sie04]), jolla kyseinen toiminnallisuus paketoidaan sovellusohjelman käyttöön, ja vielä tälle yhteyksiä korkeamman tason kielille, kuten [Sie04] pyrkii tekemään kirjoittamalla GLib-yhteensopivan kirjaston [GLI].

Tämän jälkeen voidaan toteuttaa palvelinkomponentti, joka tarjoaa verkkorajapinnan kyseiseen kirjastoon (ks. [Cha04]) joko oman protokollan (meidän tapauksessa STCCP) tai valmiin yleisen protokollan varaan ([Use], [SOA03], [COR03]), ja lopuksi voidaan toteuttaa käyttöliittymä joko suoraan kirjastoa käyttäen tai palvelinkomponenttiin nojaten. Jokainen näistä vaiheista on itsessään iso projekti, ja kokonaisuutena tämän mittakaavan projekti vaatii huomattavasti enemmän resursseja kuin Paketti2-ryhmässä oli saatavailla.

Ryhmämme olisi voinut selvittää näitä asioita projektin alussa, mutta silloin projektin tuottama lopputulos olisi ollut eri kuin projektin alussa asetetut tavoitteet. Nyt nämä asiat ovat kuitenkin tiedossa, ja mahdollinen jatkokehitys voi keskittyä tarkemmin yhteen, hal-

littavaan osa-alueeseen.

Lähteet

Cha04 Charvat, M., Snmp qos daemon, 2004. URL http://x-ray.prokon.cz/data/snmp/.

COR03 Common object request broker architecture, 2003. URL http://www.corba.org/.

FSF91 Gnu general public license, 1991. URL http://www.gnu.org/licenses/gpl.html.

FSF99 Gnu lesser general public license, 1999. URL http://www.gnu.org/licenses/lgpl.html.

GLI Glib core library. URL http://www.gtk.org/.

Mal99 Malipatna, G. V. . P., An api for linux qos support, 1999. URL http://www.ittc.ku.edu/~pramodh/courses/linux_qos/mainpage.html.

Ols02 Olshefski, D., Tc api, 2002. URL http://www-124.ibm.com/developerworks/projects/tcapi.

Pr03 Paketti-ryhmä, Määrittelydokumentti, 2003. URL http://www.cs.helsinki.fi/group/paketti/dokumentit/maarittelydokumentti.ps.

Pr04a Paketti2-ryhmä, Maintenancedocument, 2004. URL http://www.cs.helsinki.fi/group/paketti2/doc/maintenancedocument.pdf.

Pr04b Paketti2-ryhmä, Määrittelydokumentti, 2004. URL http://www.cs.helsinki.fi/group/paketti2/doc/maarittelydokumentti.pdf.

Pr04c Paketti2-ryhmä, Projektisuunnitelma, 2004. URL http://www.cs.helsinki.fi/group/paketti2/doc/projektisuunnitelma.pdf.

Pr04d Paketti2-ryhmä, Testaussuunnitelma, 2004. URL http://www.cs.helsinki.fi/group/paketti2/doc/testaussuunnitelma.pdf.

Sie04 Siemon, D., Lql-libarary homepage, 2004. URL http://www.coverfire.com/lql/.

SOA03 Soap version 1.2 part 0: Primer, 2003. URL http://www.w3.org/TR/2003/REC-soap12-part0-20030624/.

Use UserLandSoftwareInc, Xml-rpc. URL http://www.xmlrpc.com/.