HELSINGIN YLIOPISTO

TIETOJENKÄSITTELYTIETEEN LAITOS

Ohjelmistotuotantoprojekti

PAKETTI

Määrittelydokumentti

Marko Buuri
Lasse Gustafsson
Tommi Rantala
Pekka Rousu
Jaakko Saaristo

16. joulukuuta 2003

Versiohistoria

Versio	Päivämäärä	Muutokset
1.3	16.10.2003	Jäädytetty versio
1.2	15.10.2003	Suodattimet lisätty toimintokuvaukseen
1.1	6.10.2003	FTR-palaverissa 30.9.2003 havaitut virheet korjattu
1.0	29.9.2003	Ensimmäinen asiakkaalle toimitettu versio

Sisältö

1 Johdanto

Paketti-projekti kuuluu Helsingin yliopiston tietojenkäsittelytieteen laitoksen Ohjelmistotuotantoprojekti-kurssiin. Projektin tavoitteena on tuottaa ohjelmisto Linux-ytimen kaistanhallinta-asetusten hallinnointiin graafisessa muodossa. Järjestelmä tulee Helsingin yliopiston tietojenkäsittelytieteen laitoksen tutkimusryhmien käyttöön. Yliopisto julkaisee ohjelmiston GNU General Public License- [FSF91] tai GNU Lesser General Public License-lisenssin [FSF99] alaisuudessa.

1.1 Dokumentin tarkoitus

Määrittelydokumentti kiinnittää ohjelmistolle asetettavat vaatimukset. Se laaditaan yhteistyössä projektiryhmän ja asiakkaan välillä ja toimii sopimuksena tuotettavan ohjelmiston toiminnoista ja tietosisällöstä. Dokumentti palvelee sekä projektiryhmää että asiakasta varmistamalla, että ohjelmiston toiminnot

a. vastaavat asiakkaan vaatimuksia, ja

b. ovat projektiryhmän toteutettavissa.

Vaatimusten lisäksi dokumentissa kuvataan suunnittelun pääpiirteitä siinä määrin kuin ne ovat jo etukäteen tiedossa.

Projektiryhmä ja asiakas käyvät dokumentin sisällön yksityiskohtaisesti läpi FTR-palaverissa 30.9.2003, minkä jälkeen asiakas esittää ryhmälle dokumenttiin tehtävät muutokset. Kun projektiryhmä on toteuttanut muutokset, toimitetaan asiakkaalle korjattu versio hyväksymistä varten. Asiakkaan hyväksyessä korjatun version dokumentti jäädytetään, minkä jälkeen asiakkaalla ei ole oikeutta muuttaa ohjelmiston vaatimuksia. Mikäli dokumenttia joudutaan jäädytyksen jälkeen muuttamaan, eritellään muutokset suunnitteludokumenttiin.

1.2 Dokumentin rakenne

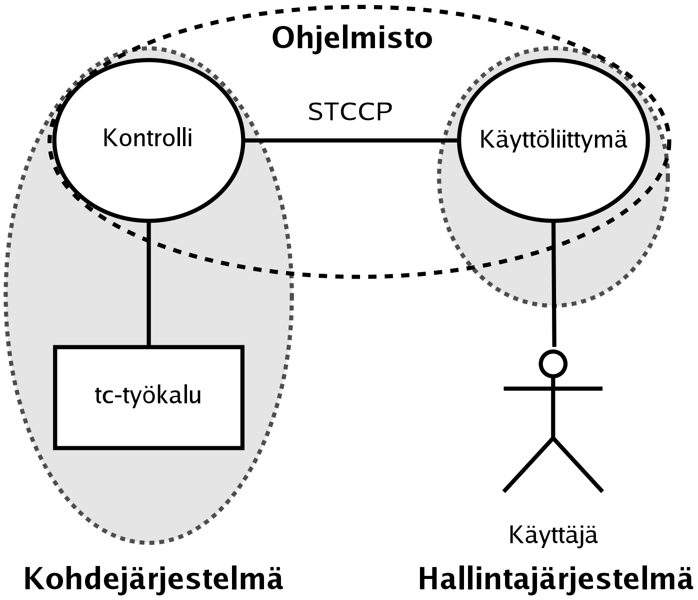
Luku 2 kuvaa tuotettavan ohjelmiston yleisen rakenteen, käyttäjät, suoritusympäristön sekä ulkoiset yhteydet. Luku 3 esittää ohjelmiston keskeiset toiminnot ja niiden potentiaaliset käyttötapaukset prioriteettijärjestyksessä. Luku 4 kuvaa ohjelmiston käsittelemän tiedon esitystapaa. Luku 5 kuvaa ohjelmiston tietoliikennetoiminnot. Luku 6 käsittää ohjelmiston muut ominaisuudet, ja viimeinen luku 7 ohjelmiston testauksen ja validoinnin.

2 Yleiskuvaus

Ohjelmiston toiminnallisuus on hajautettu kahteen pääkomponenttiin. Kontrollikomponentti tarjoaa XML-rajapinnan Linux-ytimen kaistanhallinta-asetuksiin. Käyttöliittymäkomponentti kytkeytyy tähän rajanpintaan, ja mahdollistaa kohdejärjestelmän kaistanhallinta-asetuksien käsittelyn graafisessa muodossa.

Pääkomponentit liikennöivät keskenään sovellustason protokollalla Simple Traffic Control Configuration Protocol (STCCP), jonka kuvaus on luvussa 5. Kaistanhallinta-asetukset siirtyvät XML-muotoisena komponenttien välillä.

Kuvassa 1 on ohjelmiston yleisrakenne. Ohjelmiston kontrollikomponentti on kohdejärjestelmässä, jonka kaistanhallinta-asetuksia käyttäjä muokkaa hallintajärjestelmällä.



Kuva 1: Ohjelmiston yleisrakenne

2.1 Käyttäjä

Ohjelmiston käyttäjä on Linux-järjestelmän ylläpitäjä, jonka tarvitsee muokata kaistanhallinnan asetuksia. Käyttöliittymää voi käyttää kuka tahansa käyttäjä jolla on oikeus suorittaa ohjelmia hallintajärjestelmässä. Kontrollikomponentin käyttöön tarvitaan kohdejärjestelmän pääkäyttäjän käyttöoikeus, mikäli järjestelmän kaistanhallinta-asetuksia halutaan muokata. Tämä johtuu rajoituksista Linux-ytimen rajapintojen käyttöoikeuksissa. Kontrollikomponenttia voidaan myös käyttää muun käyttäjän oikeuksilla, jolloin komponentin asetusten muunnostoiminnot ovat käytettävissä ilman asetuksien käyttöönottomahdollisuutta.

2.2 Suoritusympäristö

Ohjelmisto toteutetaan Helsingin yliopiston tietojenkäsittelytieteen laitoksen CSL 2 Linux -ympäristöön [KAL03].

Hallintajärjestelmä

Hallintajärjestelmä suorittaa ohjelmiston käyttöliittymäkomponenttia, joka toteutetaan Java-kielellä. Seuraavat komponentit ja ohjelmistot tarvitaan hallintajärjestelmässä:

- **Java 2 Runtime Environment 1.4.1** tarvitaan käyttöliittymäkomponentin suorittamiseen. **Java 2 Standard Edition SDK 1.4.1** tarvitaan komponentin kääntämiseen lähdekoodeista. Käyttöliittymä saattaa toimia tätä vanhemmilla versioilla, mutta yhteensopivuutta ei taata.
- **Xerces2**-kirjasto tai Java 2:n sisäänrakennettu XML-jäsennin tarvitaan sovelluksen kääntämiseen lähdekoodeista.
- Mikäli käyttäjä haluaa muokata muun kuin paikallisen järjestelmän kaistanhallinta-asetuksia, tarvitaan toimiva **TCP/IP-yhteys** kohdejärjestelmään.

Kohdejärjestelmä

Kohdejärjestelmä suorittaa ohjelmiston kontrollikomponenttia, joka toteutetaan C-kielellä. Seuraavat komponentit ja ohjelmiston tarvitaan kohdejärjestelmässä:

- Komponentti suunnitellaan **Linux-ytimelle 2.4.20-19.9.csl**. Ytimeen on käännetty tuki tarvittaville kaistanhallinta-ominaisuuksille.
- Komponentti käyttää GNU C-kirjastoa. Komponentin kääntämiseen ja suorittamiseen tarvitaan **glibc 2.3.2**.
- Komponentti käyttää **xinetd 2.3.11**-palvelua tietoliikenneyhteyksissä.
- Tc-työkalu **iproute 2.4.7**-paketista tarvitaan kaistanhallinta-asetusten käyttöönottoon kohdejärjestelmässä.

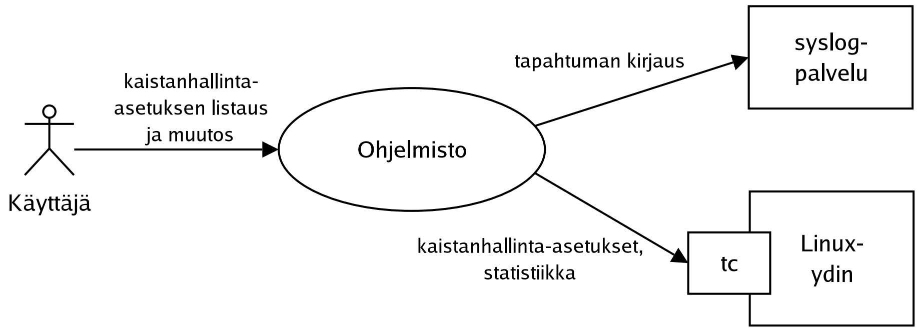
- **libxml**- ja **libxslt**-kirjastot tai **Xerces-C++**-kirjasto tarvitaan komponentin kääntämiseen lähdekoodeista.

- Valinnainen **syslog-yhteensopiva palvelu** tarvitaan lokitapahtumien kirjaamiseen.

2.3 Yhteydet

Käyttäjän lisäksi ohjelmistoon ei ole muita suunniteltuja käyttösidoksia. Ohjelmistossa on käyttöyhteydet syslog-palveluun ja tc-työkaluun. Nämä yhteydet on esitetty kuvassa 2.

Käyttäjän ohjelmistolle tekemät palvelupyynnöt ovat kaistanhallinta-asetusten ja statistiikkojen listaus, asetusten muokkaaminen, sekä asetusten käyttöönotto ja poistaminen. Ohjelmiston tekemät palvelupyynnöt syslog-palvelulle ovat lokitapahtumien kirjauspyynnöt. Ohjelmiston tekemät palvelupyynnöt tc-työkalulle ovat kaistanhallinta-asetusten ja statistiikkojen listaus, asetusten poisto sekä uusien asetusten asettaminen.



Kuva 2: Ohjelmiston ulkoiset yhteydet

2.4 Vastaavat järjestelmät

Stef Coene on ohjelmoinut useita komentotiedostoja ja työkaluja kaistanhallinta-asetusten visualisointiin ja muokkaukseen [COE01].

- **QoS configurator**-niminen WWW-pohjainen käyttöliittymä kaistanhallinta-asetusten tekoon,

- **Tcgraph** ja **Bar** Java-sovelmat statistiikan tulostamiseen,

- käytetyn kaistan ilmaiseva **monitor.pl**-komentotiedosto, sekä

- kaistanhallinta-asetuksista graafin luova **show.pl**-komentotiedosto.

Zoot-nimimerkillä esiintyvän henkilön tekemä **CBQ-based Traffic Control GUI** [ZOO03] keskittyy CBQ-jonomallin käyttöön kaistanhallinta-asetuksissa. Käyttöliittymä on ohjelmoitu käyttäen GTK-kirjastoja. Ohjelmisto on melko varhaisessa kehitysvaiheessa.

Yllä mainitut ohjelmistot eivät sovellu Paketti-projektin käyttöön laajennusmielessä, sillä ne ovat joko ohjelmoitu sellaisilla ohjelmointikielillä joita tässä projektissa ei käytetä, tai ne ovat pohjimmiltaan liian rajoittuneita haluttuun yleiskäyttöisyyteen.

3 Toimintokuvaus

Seuraavassa luetellaan ohjelmiston toiminnot ja esitellään niihin liittyviä käyttötapauksia. Jokaisesta toiminnosta on listattu koodi, kuvaus ja prioriteetti.

Prioriteetin 1 toiminnot toteutetaan, ja näiden toimintojen toimivuus on vaatimus projektin onnistumiselle.

Prioriteetin 2 toiminnot toteutetaan, mikäli prioriteetin 1 toiminnot todetaan testauksen kautta toimiviksi, ja toteutusvaiheessa on vielä aikaa jäljellä.

Prioriteetin 3 toimintoja ei todennäköisesti toteuteta.

3.1 Käyttöliittymän toiminnot

koodi	**kuvaus**	**pr**
KL1	Kaistanhallinta-asetusten esitys graafisessa muodossa.	1
KL2	Kaistanhallinta-asetusten luominen graafisessa muodossa.	1
KL3	Kaistanhallinta-asetusten muokkaus graafisessa muodossa.	1
KL4	Kaistanhallinta-asetusten esitys tc-komentoina.	1
KL5	STCCP-protokollan toteuttaminen.	1
KL6	Kohdejärjestelmän kontrollikomponenttiin kytkeytyminen TCP/IP-yhteydellä.	1
KL7	Kohdejärjestelmän kaistanhallinta-asetusten haku.	1
KL8	Kohdejärjestelmän kaistanhallinta-asetusten asettaminen.	1
KL9	Kohdejärjestelmän kaistanhallinnan tilastotietojen tarkastelu tekstimuodossa.	1

KL10	Alustustietojen lukeminen tiedostosta.	2
KL11	Käyttäjän ohjeistus asetuksien määrittelyissä.	2
KL12	Kelvottomien kaistanhallinta-asetusten luonnin estäminen.	2
KL13	Kaistanhallinta-asetusten tallennus paikalliseen tiedostojärjestelmään XML-muodossa.	2
KL14	Kaistanhallinta-asetusten lataus paikallisesta tiedostojärjestelmästä XML-muodossa.	2
KL15	Kaistanhallinta-asetusten tallennus paikalliseen tiedostojärjestelmään tc-komentotiedostona kontrollikomponentin avulla.	2
KL16	Kaistanhallinta-asetusten lataus paikallisesta tiedostojärjestelmästä tc-komentotiedostomuodossa kontrollikomponentin avulla.	2
KL17	Kohdejärjestelmän kaistanhallinnan tilastotietojen tallennus paikalliseen tiedostojärjestelmään tekstimuodossa.	2
KL18	Tuki kieliversioille.	2
KL19	Käytettävissä useampi kuin yksi kieliversio.	3
KL20	Kohdejärjestelmän kaistanhallinnan tilastotietojen tarkastelu graafisessa muodossa.	3
KL21	Kaistanhallinta-asetusten graafisen esitysmuodon tulostus.	3
KL22	Kirjanpito aiemmin avatuista hallintayhteyksistä.	3
KL23	Kirjanpito viimeksi käsitellyistä tiedostoista.	3
KL24	Valmiita esimerkkejä kaistanhallinta-asetuksista.	3

3.2 Kontrollikomponentin toiminnot

koodi	**kuvaus**	**pr**
KK1	STCCP-protokollan toteuttaminen.	1
KK2	Kaistanhallinta-asetusten XML-muodon muuntaminen tc-komennoiksi.	1
KK3	Tc-komentojen suoritus.	1

KK4	Kaistanhallinta-asetusten lähettäminen käyttöliittymälle XML-muodossa.	1
KK5	Kaistanhallinta-asetusten lähettäminen käyttöliittymälle tc-komentotiedostomuodossa.	1
KK6	Tc-työkalun kaistanhallinta-asetusten tulosteiden tulkitseminen ja muuntaminen XML-muotoon.	1
KK7	Kaistanhallinnan tilastotietojen nouto tc-työkalulla.	1
KK8	Tc-komentojen muuntaminen XML-muotoon.	1
KK9	Lokitietojen tulostaminen syslog-palveluun.	1
KK10	Alustustietojen lukeminen tiedostosta.	2
KK11	Yhtäaikaisten hallintayhteyksien poissulkeminen.	3

3.3 Ohjelmiston tukemat jonomallit

Jonomallien toiminnan esittelee Liite 3: Jonomallit.

koodi	**kuvaus**	**pr**
J1	Hierarchical Token Bucket, HTB	1
J2	Packet limited fifo, pfifo	1
J3	Byte limited fifo, bfifo	1
J4	Random Early Detect, RED	1
J5	Priority, PRIO	2
J6	Token Bucket Filter, TBF	2
J7	Stochastic Fairness Queueing, SFQ	2
J8	Class Based Queueing, CBQ	3
J9	Differentiated Services, DSMARK	3
J10	Weighted Fair Queueing, WFQ	3

3.4 Ohjelmiston tukemat suodattimet

koodi	**kuvaus**	**pr**
S1	u32	1
S2	fw	2
S3	route	2
S4	tcindex	2

3.5 Tietoliikenneprotokollan toiminnot

koodi	**kuvaus**	**pr**
TL1	Viestien autentikointi digitaalisilla allerkirjoituksilla.	2
TL2	Tietoliikenneviestien tiivistäminen.	3

3.6 Käyttötapaukset

Seuraavassa on lueteltu ohjelmiston käyttötapauksia. Suoritettavan toiminnan yhteydessä on lueteltu käyttötapauksen kannalta oleellisimpien toimintojen koodit.

Käyttötapaus KT1: *Kohdejärjestelmän kaistanhallinta-asetusten tarkastelu.*
Tavoite: Tarkastella kohdejärjestelmän kaistanhallinta-asetuksia.
Toiminta: Käyttäjä käynnistää käyttöliittymän, joka ottaa yhteyden kohdejärjestelmään (KL5, KL6, KK1, KK9, KK10). Käyttöliittymä pyytää kontrollikomponentilta senhetkiset kaistanhallinta-asetukset (KL7, KK3, KK4, KK6). Tämän jälkeen asetukset ovat tarkasteltavissa sekä visuaalisessa (KL1), että tc-komentomuodossa (KL4, KK2).

Käyttötapaus KT2: *Kohdejärjestelmän kaistanhallinnan tilastotietojen tarkastelu.*
Tavoite: Kaistanhallinta-asetusten toimivuuden varmistaminen.
Toiminta: Käyttäjä painaa tilastotiedot näyttävää nappia (KL9). Käyttöliittymä hakee tiedot kontrollikomponentilta (KK7).

Käyttötapaus KT3: *Kaistanhallinta-asetusten käyttöönotto.*
Tavoite: Kaistanhallinta-asetusten käyttöönotto.

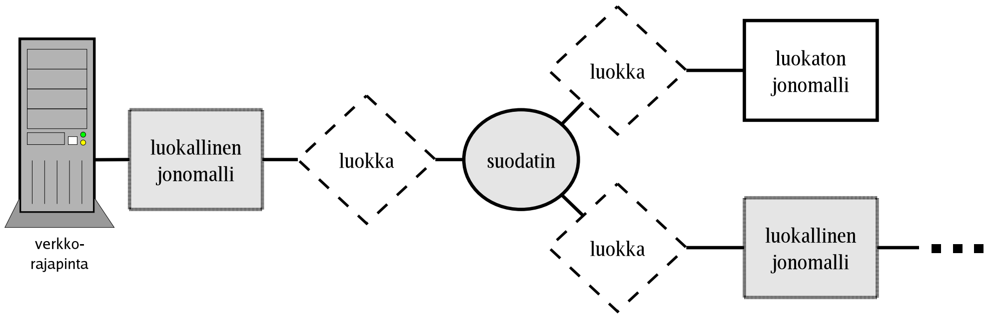
Toiminta: Käyttäjä painaa asetukset käyttöönottavaa nappia (KL8). Kontrollikomponentti asettaa uudet asetukset voimaan (KK2, KK3).

Käyttötapaus KT4: *Tc-komentotiedostojen esitys graafisessa muodossa.*
Tavoite: Valmiiden tc-komentotiedostojen esitys ja muokkaus käyttöliittymässä.
Toiminta: Käyttäjä painaa tc-komentotiedoston avausnappia ja antaa syötteenä avattavan tiedoston (KL16, KK8).

Käyttötapaus KT5: *Kaistanhallinta-asetusten tallennus tc-komentomuodossa.*
Tavoite: Käyttäjä haluaa ottaa laatimansa kaistanhallinta-asetukset käyttöön oletusasetuksina.
Toiminta: Käyttäjä painaa tc-komentotiedostoon tallentamis -nappia ja tallennuksen jälkeen kopioi tc-komentotiedoston kohdejärjestelmän käynnistystiedostojen hakemistoon. (KK2, KK5)

4 Tietokuvaus

Kaistanhallinta-asetukset koostuvat jonomalleista (Liite 3: Jonomallit), luokista sekä suodattimista (Kuva 3). Jonomalleja on sekä luokallisia että luokattomia, ja ne määrittelevät IP-pakettien lähetysjärjestyksen. Jokaisessa laiterajapinnassa on aina jokin jonomalli. Luokilla voidaan muodostaa puumainen hierarkia, jota pitkin paketit etenevät suodattimien määräämillä säännöillä. Jokainen lehtiluokka sisältää jonomallin. Suodattimia tarvitaan luokittelemaan paketteja eri luokkiin.



Kuva 3: Kuvaus kaistanhallinta-asetuksista

4.1 Asetuksien esitysmuoto

Käyttöliittymäkomponentti käsittelee kaistanhallinta-asetuksia Java-luokkina. Jonomallit, luokat ja suodattimet mallinnetaan olioina, joiden kentät sisältävät näiden asetukset. Kontrollikomponentti käsittelee asetuksia linkitetyssä tietorakenteessa. Jonomallien, luokkien ja suotimien asetukset tallennetaan linkitettyjen tietueiden kenttiin. Tarkat tietorakenteiden kuvaukset valmistuvat projektin suunnitteludokumenttiin.

Kaistanhallinnan asetukset siirretään ohjelmiston pääkomponenttien välillä XML-muotoisina. XML-dokumentin rakenne kuvataan joko XML Document Type Definition- (DTD) [W3C00] tai XML Schema-menetelmällä [W3C03]. Kaikki hallittavan kohdejärjestelmän tieto, jota on tarpeen ohjelmiston komponenttien välillä siirtää, kuvataan yhdessä dokumentissa.

XML-rakenteen kuvaus valmistuu projektin suunnitteludokumenttiin. Esimerkki mahdollisesta siirrossa käytetystä XML-dokumentista on seuraava:

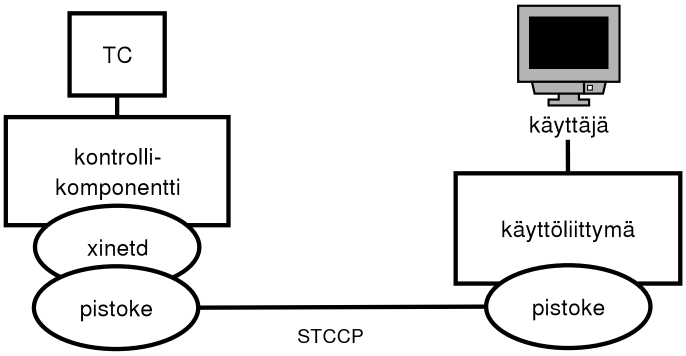
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<system>
  <hostname>...</hostname>
  <domain>...</domain>
  <defaultgatewayip>...</defaultgatewayip>
  <defaultgatewayinterface>...</defaultgatewayinterface>
  <interface>
    <name>...</name>
    <rate>...</rate>
    <ip>...</ip>
    <subnetmask>...</subnetmask>
    <up>...</up>
    <rootqdisc>
      ...
    </rootqdisc>
  </interface>
  <interface>
    ...
  </interface>
  ...
</system>
```

4.2 Alustustiedot

Sekä käyttöliittymän että kontrollikomponentin alustustiedot syötetään komentoriviparametreina. Mikäli toiminto TL1 toteutetaan, luodaan ohjelmiston osille alustustiedostot, minne tallennetaan autentikoinnissa käytettävä tunnus ja salasana.

5 Tietoliikenne

Ohjelmiston tietoliikenne tapahtuu kontrollikomponentin ja käyttöliittymän välillä. Yleiskuva tietoliikenteestä on esitetty kuvassa 4.



Kuva 4: Ohjelmiston tietoliikenne

5.1 Yhteystapa

Tietoliikenneyhteyden muodostamiseen käytetään TCP/IP-protokollaa. Kontrollikomponentissa käytetään xinetd-palvelua ja käyttöliittymässä Javan tarjoamia pistokkeita. Kun käyttöliittymä käynnistetään, se ottaa yhteyden kontrollikomponenttiin pistokkeen kautta. Viestin vastaanottaa xinetd-palvelu, joka aluksi käynnistää kontrollikomponentin, minkä jälkeen se ohjaa pistokkeeseen saapuvan tietoliikenteen kontrollikomponentin standardisyötevirtaan ja tältä tulevan standarditulostusvirran pistokkeeseen. Käyttöliittymä lähettää lopuksi sulkemispyynnön, joka katkaisee yhteyden ja lopettaa kontrollikomponentin suorittamisen.

5.2 Protokolla

Projekti määrittelee tietoliikennettä varten oman sovellustason protokollan Simple Traffic Control Configuration Protocol (STCCP), sillä valmiita sopivia protokollia ei ollut. Protokollaa käytetään siirtämään tietoa kontrollikomponentin ja käyttöliittymän välillä. Protokollan tarkka kuvaus liitetään suunnitteludokumenttiin, mutta seuraavassa hahmotellaan sen pääpiirteitä.

STCCP-viestin sisältö kuvataan XML-muotoisena, joko XML Document Type Definition- (DTD) [W3C00] tai XML Schema-menetelmällä [W3C03]. Viestin pakollisia elementtejä ovat protokollan versionumero, viestin järjestysnumero, sekä komento. Valinnaisia elementtejä ovat siirrettävä tietosisältö sekä HMAC-kenttä [KBC97] tai vastaava viestien autentikointia varten. Versionumerolla varmistetaan kontrollikomponentin ja käyttöliittymän olevan yhteensopivia. Järjestysnumeroilla varmistetaan viestien prosessointi oikeassa järjestyksessä. Käytetyt komennot ovat seuraavanlaisia:

GET pyytää kontrollikomponentilta kaistanhallinta-asetuksia tai statistiikkaa liikenteestä.

TRANSLATE pyytää kontrollikomponenttia tekemään käännöksen asetuksista XML-muotoon tai tc-komentomuotoon. Asetukset tulevat osana viestiä.

SET käskee kontrollikomponenttia tekemään annetun asetuksen mukaiset muutokset kaistanhallintaan. Asetukset tulevat osana viestiä.

QUIT käskee kontrollikomponenttia katkaisemaan yhteyden.

HELLO varmistaa yhteyden muodostumisen kontrollikomponentin ja käyttöliittymän välillä.

ERROR ilmoittaa tapahtuneesta virhetilanteesta.

REPLY komento ilmaisee vastauksen pyyntökomentoihin.

Esimerkki mahdollista lähetteen DTD-kuvauksesta:

```
<!ELEMENT MESSAGE (VERSION, SEQUENCE, COMMAND, DATA?, HMAC?)>
<!ELEMENT VERSION(#PCDATA)>
<!ELEMENT SEQUENCE(#PCDATA)>
<!ELEMENT COMMAND(#PCDATA)>
<!ELEMENT DATA(#CDATA)>
<!ELEMENT HMAC(#PCDATA)>
```

5.3 Sekvenssikaavio

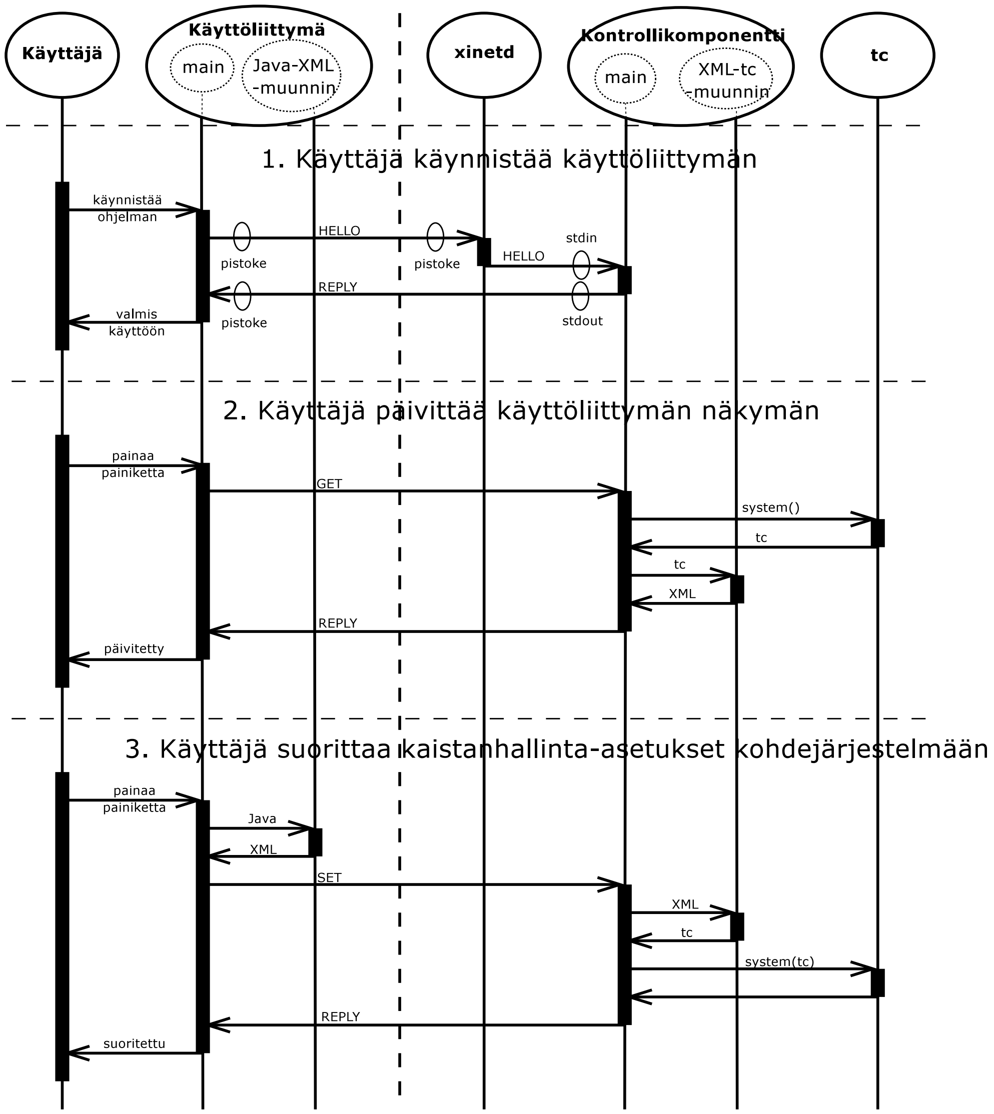
Ohjelmiston tietoliikennettä kuvaava sekvenssikaavio on esitetty kuvassa 5.

6 Muut ominaisuudet

Tässä luvussa käsitellään muita ohjelmistolta vaadittavia ominaisuuksia: suorituskykyä, käytettävyyttä, tietoturvaa sekä ylläpidettävyyttä.

6.1 Suorituskyky

Ohjelmiston käyttöliittymän ja kontrolliosan algoritmit sekä tietorakenteet suunnitellaan ajan ja tilan suhteen tehokkaiksi. Koska ohjelmistoa tulisi käyttää yksi käyttäjä kerrallaan, ei kontrollikomponentin suorituskyky ole kriittinen. Ohjelmiston tiedonsiirtotarve on vähäinen ja siirrettävät datamäärät pieniä. Olennainen käytettävyyteen vaikuttava asia on kaistanhallintatietojen lataamisen ja tallentamisen vasteaika. Koska tietoliikenne TCP/IP-yhteyden yli voi olla hidasta, toteutetaan käyttöliittymä siten, että tietojen lataaminen ja tallettaminen suoritetaan erillisestä painikkeesta.



Kuva 5: Ohjelmiston sisäinen tietoliikenne eri toiminnoilla

6.2 Käytettävyys

Ohjelmiston suorituksen aikana tapahtuvista virhetilanteista ilmoitetaan käyttäjälle. Virhetilanteita voivat aiheuttaa häiriöt tietoliikenteessä sekä tiedostoista luetut virheelliset kaistanhallinta-asetukset.

6.3 Tietoturva

Ohjelmisto voi aiheuttaa vakavia tietoturvariskejä ellei sen käyttöä ole tarkasti rajoitettu luotettaville henkilöille. Sovellustason protokolla välittää tietoa kohdejärjestelmän verkkoliityntöjen IP- sekä kaistanhallinta-asetuksista. Sovellustason protokollaa ei salata, joten ohjelmistossa käytettävän informaation kaappaaminen ulkopuoliselle henkilölle on mahdollista.

STCCP-protokollaan määritellään tapa autentikoida lähetteet käyttäen lähetekohtaista HMAC-allekirjoitusta [KBC97], jota käytetään myös esimerkiksi IPSec- ja SSL/TLS-prokollissa. HMAC-menetelmä määrittelee tavan, kuinka viestin sisällöstä lasketaan valinnaisen hajautusfunktion sekä salaisen avaimen avulla digitaalinen allekirjoitus. Salainen avain on tallennettu sekä kohde- että hallintajärjestelmiin. HMAC-toiminto toteutetaan ohjelmistoon toiminnossa TL1, jossa hajautusfunktiona käytetään joko MD5- [RRI92] tai SHA-1-menetelmää [EJO01].

Ohjelmistoon toteutetaan mahdollisuus hallintayhteyden pääsyoikeuden rajoittamiseen salasanalla.

6.4 Ylläpidettävyys

Ohjelmiston tuotannossa pyritään selkeään ohjelmointityyliin ja lähdekoodin luettavuutta ylläpidetään kattavalla kommentoinnilla. Ohjelmiston suunnittelussa on huomioitu mahdollinen tulevaisuuden intressi korvata jompikumpi ohjelmiston komponenteista toisella. Komponenttien välinen lopullinen rajapinta kuvataan suunnitteludokumenttiin. Ylläpidettävyyteen liittyvät asiat dokumentoidaan ylläpitodokumenttiin.

7 Testaus

Projektiryhmä testaa ohjelmiston projektin testausvaiheessa, joka alkaa toteutusvaiheen aikana ja päättyy toteutusvaiheen jälkeen. Tarkka testaussuunnitelma valmistuu omana dokumenttinaan ennen testausvaiheen alkamista. Testaus suoritetaan vaiheittain yksikkötestauksena, integraatiotestauksena, sekä validointitestauksena. Ryhmä korjaa testausvaiheissa löytyneet virheet, mikäli ohjelmiston toteutusvaihe ei ole päättynyt.

7.1 Yksikkötestaus

Ryhmä suorittaa yksikkötestauksen jokaiselle ohjelmiston osakomponentille. Testauksen tehtävänä on varmistaa yksittäisten komponenttien toimivuus dokumentoidulla tavalla. Valtaosa testeistä tehdään testaamalla komponenttien toimivuutta dokumentoitujen rajapintojen kautta black-box-testauksena. Aikataulun salliessa ohjelmistolle tehdään myös white-box -testauksia.

7.2 Integraatiotestaus

Ryhmä suorittaa integraatiotestauksen useiden komponenttien joukoille dokumentoitujen rajapintojen kautta black-box-testauksena. Testauksen tavoitteena on varmistaa komponenttien keskinäinen yhteentoimivuus ja dokumentoida mahdolliset virheet. Löytyneet virheet korjataan, mikäli ohjelmiston toteutusvaihe ei ole päättynyt. Integraatiotestaus tehdään vähintään kontrolli- ja käyttöliittymäkomponenteille omina kokonaisuuksinaan.

7.3 Validointitestaus

Validointitesteissä etsitään ja dokumentoidaan suunnittelun puutteita ja virheitä, kun ohjelmiston toteutusvaihe on päättynyt. Ohjelmiston osalta projekti on onnistunut, kun prioriteetin 1 toiminnot toimivat määrittelyn mukaisesti ja ohjelmakoodi on helposti ylläpidettävää.

Viitteet

[COE01] Coene, S., Kaistanhallintaa käsittelevä sivusto, 2001. http://www.docum.org,1.10.2003

[EJO01] Eastlake, D. ja Jones, P., RFC 3174 - US Secure Hash Algorithm 1 (SHA1), 2001. http://www.ietf.org/rfc/rfc3174.txt,4.10.2003

[FSF91] Free Software Foundation, Inc., GNU General Public License, 1991. http://www.gnu.org/copyleft/gpl.html,1.10.2003

[FSF99] Free Software Foundation, Inc., GNU Lesser General Public License, 1999. http://www.gnu.org/copyleft/lesser.html,1.10.2003

[HUB00] Hubert, Bert et.al., Linux 2.4 Advanced Routing HOWTO, 2000. http://www.linuxdocs.org/HOWTOs/Adv-Routing-HOWTO.html,1.10.2003

[KAL03] Kalliokoski, P., CSL-ympäristön erityisominaisuudet, 2003. http://www.ling.helsinki.fi/atk/admin/csl_features.xhtml,14.9.2003

[KBC97] Krawczyk, H., Bellare, M. ja Canetti, R., RFC 2104 - HMAC: Keyed-Hashing for Message Authentication, 1997. http://www.ietf.org/rfc/rfc2104.txt,28.9.2003

[RRI92] Rivest, R., RFC 1321 - The MD5 Message-Digest Algorithm, 1992. http://www.ietf.org/rfc/rfc1321.txt,4.10.2003

[W3C00] World Wide Web Consortium, Extensible Markup Language (XML) 1.0, 2000. http://www.w3.org/TR/REC-xml,28.9.2003

[W3C03] World Wide Web Consortium, XML Schema, 2003. http://www.w3.org/XML/Schema,28.9.2003

[ZOO03] Zoot, CBQ-based Traffic Control GUI, 2003. http://users.skynet.be/cbqinit,27.9.2003

Liite 1: Termistö

Seuraavassa annetaan selitykset Paketti-projektin dokumenteissa käytetyille termeille.

asiakas palvelun käyttäjä, esimerkiksi käyttöliittymä

hallintajärjestelmä järjestelmä, jolta käyttöliittymää käytetään.

HMAC *Keyed-Hashing for Message Authentication* [KBC97], kryptograafisesti vahva menetelmä viestien autentikointiin.

jono pakettijono; kaistanhallinnassa paketit sijoitetaan jonoihin, joiden toimintaa säännöstelee jonomalli [HUB00].

jonomalli kaistanhallinnan väline; algoritmi, joka säännöstelee jonoa (engl. *queuing discipline, qdisc*) [HUB00].

kaistanhallinta tietoliikenteen hallinta laatutakuiden toteuttamiseksi (engl. *Bandwidth management*).

kohdejärjestelmä järjestelmä, jonka kaistanhallinta-asetuksia sovelluksella muutetaan.

kontrollikomponentti ohjelmiston palvelinosa, joka tarjoaa rajapinnan kaistanhallinta-asetuksiin.

käyttöliittymä ohjelmiston graafinen käyttöliittymä.

käyttöliittymäkomponentti ohjelmiston asiakasosa, joka sisältää käyttöliittymän (vrt. kontrollikomponentti)

laatutakuu asetusten kokonaisuus, jolla pyritään parantamaan jonkun palvelun käytettävyyttä tietoliikenneverkossa (engl. *Quality of Service, QoS*).

Linux GNU/Linux-käyttöjärjestelmä.

ohjelmisto Paketti-projektin tuottama ohjelmisto kaistanhallinta-asetusten määrittelyyn graafisesti.

paketti käyttöliittymäkomponentin binääritiedosto (vrt. Paketti-projekti isolla alkukirjaimella)

pakettid	kontrollikomponentin binääritiedosto
palvelin	palvelinohjelmistoa suorittava järjestelmä
pyyntö	STCCP-viesti, jolla asiakas tekee palvelimelle palvelupyynnön
pääkäyttäjä	Linux-järjestelmän pääkäyttäjä (root).
STCCP	*Simple Traffic Control Configuration Protocol.* Projektin määrittelemä protokolla kontrollikomponentin ja käyttöliittymän väliseen tiedonsiirtoon.
STCCP-viesti	STCCP-protokollan mukainen yksittäinen viesti, joko pyyntö tai vastaus
suodatin	kaistanhallinnan väline IP-pakettien jakamiseen luokkiin (engl. *filter*).
vastaus	STCCP-viesti, jolla palvelin vastaa asiakkaan pyyntöön.
ydin	Linux-järjestelmän ydin (engl. *kernel*).
XSL	*Extensible Stylesheet Language*
XSLT	*XSL Transformations*

Liite 2: Tc-ohjelma

Tc on komentotyökalu jolla voidaan muuttaa ytimen kaistanhallinta-asetuksia. Yksittäinen komento koskee jonomallia, luokkaa, tai suodatinta, joita voidaan lisätä, poistaa ja muokata. Tc-ohjelma osaa listata käytössä olevat asetukset, sekä kertoa tilastotietoa jonojen kautta kulkeneesta liikenteestä ja tehdyistä rajoitustoimista.

Tc-ohjelman syntaksi on seuraava:

```
tc qdisc [ add | del | replace | change | get ]
   dev STRING [ handle QHANDLE ] [ root | ingress | parent CLASSID ]
   [ estimator INTERVAL TIME_CONSTANT ]
   [ [ QDISC_KIND ] [ help | OPTIONS ] ]

tc qdisc show [ dev STRING ] [ingress]

tc class [ add | del | change | get ]
   dev STRING [ classid CLASSID ] [ root | parent CLASSID ]
   [ [ QDISC_KIND ] [ help | OPTIONS ] ]

tc class show [ dev STRING ] [ root | parent CLASSID ]

tc filter [ add | del | change | get ]
   dev STRING [ pref PRIO ] [ protocol PROTO ]
   [ estimator INTERVAL TIME_CONSTANT ]
   [ root | classid CLASSID ] [ handle FILTERID ]
   [ [ FILTER_TYPE ] [ help | OPTIONS ] ]

tc filter show [ dev STRING ] [ root | parent CLASSID ]
```

Liite 3: Jonomallit

Jonomallit jaetaan kahteen tyyppiin: luokattomiin ja luokallisiin. Luokalliset jonomallit sisältävät halutun määrän luokkia, joihin voidaan kiinnittää suodattimia ja uusia jonomalleja. Luokkien avulla asetukset muodostavat puurakenteen.

Luokattomat jonomallit

PFIFO_FAST on oletuksena käytössä jokaisessa laitteistorajapinnassa mikäli järjestelmän ydin on käännetty Advanced Router-optiolla. Se luo kolme FIFO-jonoa, joista ensimmäisellä on suurin ja viimeisellä pienin prioriteetti lähettää IP-paketteja eteenpäin. Paketit jaetaan jonoihin niiden prioriteettibittien mukaan.

PFIFO (Packet Based First In First Out) ja **BFIFO** (Byte Based First In First Out) ovat IP-paketti- tai tavuperustaiset jonot. Jonon koko on määriteltävissä.

RED (Random Early Detection) simuloi ruuhkaista reititintä, joka pudottaa satunnaisesti IP-paketteja kun kaista uhkaa loppua. Tämä estää todellisen ruuhkatilanteen syntymisen, jolloin kaikki TCP-yhteydet hidastuisivat. Pudotettavaa pakettia ei valita aivan satunnaisesti vaan todennäköisemmin valittavaksi joutuu paljon paketteja lähettävän yhteyden paketti. RED soveltuukin paljon liikennettä sisältäviin, lähes tukkoisiin kohteisiin.

TBF (Token Bucket Filter) on jonomalli, jolla voidaan rajoittaa kaistanleveyden käyttö määriteltyyn ylärajaan. TBF sallii lyhytaikaiset maksimimäärän ylittävät purskeet.

DSMARK tarjoaa resurssit, joita tarvitaan Differentiated Services-palvelussa. Differentiated Services on toinen kaistanhallinta-arkkitehtuuri, joka perustuu IP-pakettien otsakkeissa olevaan DS-kenttään.

Luokalliset jonomallit

HTB (Hierarchical Token Bucket) mahdollistaa luokkien kaistanleveyden käytön rajoittamisen.

CBQ (Class Based Queueing) mahdollistaa luokkien kaistanleveyden käytön rajoittamisen. Sisäiseltä toiminnaltaan CBQ kuitenkin eroaa HTB-jonomallista.

PRIO (Priority) on FIFO:n tyyppinen jonomalli mutta se luo kaistojen sijasta luokkia. IP-paketit jaetaan eri luokkiin määritellyn suodattimen mukaan. Jokainen luokka käyttää oletuksena FIFO-jonomallia mutta sen voi vaihtaa haluamaansa jonomalliin. PRIO on hyödyllinen kun paketteja halutaan jakaa käyttämällä myös suodatinta eikä pelkästään prioriteettibittejä.