HELSINGIN YLIOPISTO
TIETOJENKÄSITTELYTIETEEN LAITOS

Ohjelmistotuotantoprojekti

PAKETTI

Suunnitteludokumentti

Marko Buuri
Lasse Gustafsson
Tommi Rantala
Pekka Rousu
Jaakko Saaristo

16. joulukuuta 2003

Versiohistoria

Versio	Päivämäärä	Muutokset
1.3	6.11.2003	Jäädytetty versio
1.2	4.11.2003	FTR-palaverissa havaitut puutteet korjaukset
1.1	27.10.2003	FTR-palaverissa käsiteltävä versio
1.0	20.10.2003	Asiakkaalle toimitettu protokollan ja käyttöliittymän kuvaus

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

Suunnitteludokumenttiin on koottu suunnitteluvaiheessa tehdyt ohjelman toteutusta koskevat päätökset. Dokumentti toimii projektiryhmän ohjenuorana toteutusvaiheen aikana. Dokumentissa esitetään tuotettavan ohjelmiston rakenne sillä tarkkuudella, että toteutus on dokumentin pohjalta suoraviivaista.

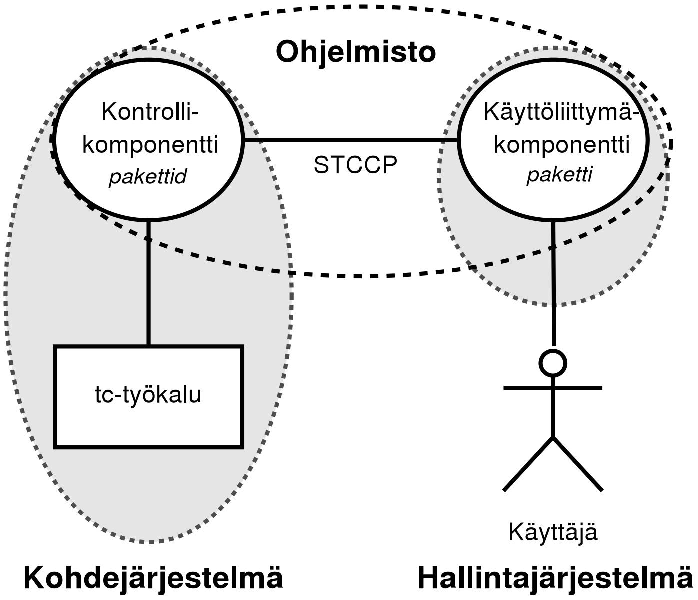
Suunnittelussa on otettu huomioon määrittelydokumentin luvussa 3 listatuista toiminnoista prioriteetille 1 merkityt toiminnot [PAK03a]. Prioriteettien 2 ja 3 toimintojen toteuttamisesta päätetään aikataulun salliessa toteutusvaiheen aikana.

1.2 Dokumentin rakenne

Luvussa 2 esitellään ohjelmiston yleisarkkitehtuuri, luvussa 3 ohjelmiston käyttämät tietorakenteet ja luvussa 4 ohjelmiston käyttämä tiedonsiirtoprotokolla. Ohjelmiston kahden pääkomponentin toiminnan tarkempi selittäminen on jaettu lukuihin 5 ja 6. Lopussa on luettelot dokumentin lähdeviitteistä, termeistä, kuvista ja taulukoista.

2 Ohjelmisto

Sovellus on hajautettu kahteen osaan, käyttöliittymäkomponentti pakettiin ja kontrollikomponentti pakettid:hen, jotka voivat sijaita eri järjestelmissä (kuva 1). Sovelluksen käyttöliittymä sijaitsee käyttöliittymäkomponentissa, joka toteutetaan Java-kielellä ja Swing-käyttöliittymäkirjastolla. Kaistanhallinta-asetusten käyttöönotosta vastaa kohdejärjestelmässä sijaitseva kontrollikomponentti tc-komentotyökalun avulla. Kontrollikomponentti toteutetaan C-kielellä. Komponenttien välistä tietoliikennettä varten projekti on määritellyt STCCP-protokollan, joka toimii TCP/IP-protokollan päällä.



Kuva 1: Ohjelmiston arkkitehtuuri

2.1 Ohjelmiston toiminnot

Tässä luvussa kuvataan sellaiset ohjelmiston toiminnot, jotka tarvitsevat käyttöliittymä- ja kontrollikomponentin välistä viestinvälitystä STCCP-protokollalla. Tarkemmat kuvaukset löytyvät luvuista 5 ja 6.3.

Käynnistyessään käyttöliittymäkomponentti avaa TCP/IP-yhteyden kohdejärjestelmän kontrollikomponenttiin. Yhteys pysyy auki, kunnes käyttöliittymä suljetaan.

Kaikki ohjelmiston toiminnot on listattu määrittelydokumentin luvussa 3 [PAK03a].

2.1.1 Kaistanhallinta-asetusten nouto kohdejärjestelmästä

Käyttöliittymäkomponentti noutaa käytössä olevat kaistanhallinta-asetukset kohdejärjestelmästä ohjelman käynnistyessä, asetusten käyttöönoton yhteydessä sekä erikseen käyttäjän käskemänä.

Asetuksien noudossa käyttöliittymäkomponentti lähettää pyynnön kontrollikomponentille, joka hakee asetukset tc-työkalulla ja lähettää ne vastauksessa käyttöliittymäkomponentille. Käyttöliittymä päivittää oman näkymänsä asetuksia vastaavaksi.

2.1.2 Kaistanhallinta-asetusten nouto tc-komentomuodossa

Käyttäjän pyytäessä kaistanhallinta-asetukset tc-komentomuodossa käyttöliittymäkomponentti lähettää vastaavan pyynnön kontrollikomponentille. Kontrollikomponentti hakee asetukset tc-työkalulla ja lähettää ne vastauksessa käyttöliittymäkomponentille. Käyttöliittymä näyttää tc-komennot käyttäjälle.

2.1.3 Kaistanhallinta-asetusten käyttöönotto

Käyttäjän ottaessa muokkaamansa kaistanhallinta-asetukset käyttöön käyttöliittymäkomponentti lähettää asetukset kontrollikomponentille, joka muokkaa ne tc-komennoiksi ja suorittaa komennot. Lopuksi kontrollikomponentti hakee asetukset tc-työkalulla ja lähettää ne vastauksessa käyttöliittymäkomponentille. Käyttöliittymä päivittää oman näkymänsä asetuksia vastaavaksi.

2.1.4 Kaistanhallinnan tilastotietojen tarkastelu

Käyttäjän pyytäessä kohdejärjestelmän kaistanhallinnan tilastotietoja käyttöliittymäkomponentti lähettää pyynnön kontrollikomponentille. Kontrollikomponentti hakee tilastotiedot tc-työkalulla ja lähettää ne vastausviestissä käyttöliittymäkomponentille. Käyttöliittymä näyttää tiedot käyttäjälle.

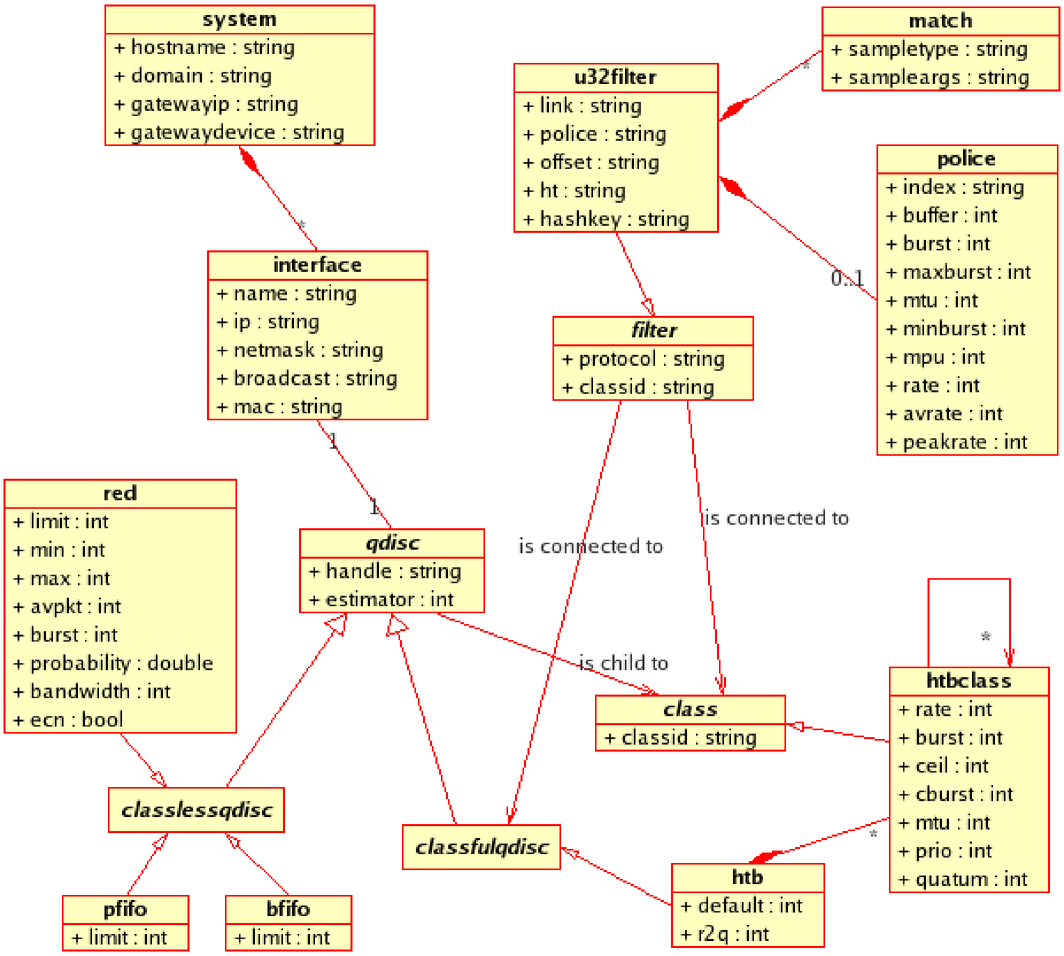
3 Tietosuunnitelma

Tässä luvussa esitellään projektin käyttämä kaistanhallinta-asetusten UML-malli sekä mallista johdettu XML-tietorakenne.

3.1 Kaistanhallinta-asetusten UML-malli

Järjestelmän kaistaistanhallinta-asetukset on kuvattu UML-rakenteena. Ohjelmiston kontrolli- ja käyttöliittymäkomponenttien väliseen kaistanhallinta--asetusten siirtoon käytetään XML-tietorakennetta, joka on johdettu UML-mallista. Komponentit käsittelevät näitä asetuksia sisäisessä toiminnassaan eri tavoin. Kuva 2 esittää kaistanhallinta-asetusten UML-mallin. Kuvassa esiintyvät ne elementit, joita projektin tuottama ohjelmisto tukee.

Mallin **System**-luokka on rakenteen juuri, johon liittyy valinnainen määrä **interface**-luokkia. Interface-luokat ilmentävät kohdejärjestelmän loogisia Ethernet-verkkoliityntöjä jotka nimetään Linux-järjestelmässä eth([0-9]|([1-9][0-9]+)), esimerkiksi "eth0" tai "eth10". Jokaiseen tällaiseen käyttöön mää-



Kuva 2: Kaistanhallinta-asetusten UML-malli

riteltyyn liityntään on liitetty yksi jonomalli. Mikäli tämä jonomalli on HTB, voidaan siihen liittää HTB-luokkia. Luokkiin voidaan edelleen liittää uusia luokki, tai jonomalleja. Suodattimia voidaan liittää sekä luokallisiin jonomalleihin että luokkiin.

U32-suodattimen valinnaisen tc-komentoparametrin selector sisältö kuvataan omina match-luokkinaan. Samoin valinnaisen police-luokan sisältö kuvataan omana luokkanaan police.

Kontrollikomponentti kääntää vastaanottamansa asetukset tc-komentomuotoon XSLT-muunnoksella. Asetuksia ei siten tallenneta erityisessä sisäisessä tietorakenteessa. Kontrollikomponentti muuntaa järjestelmän kaistanhallinta-asetukset XML-muotoon tc-työkalun tulosteiden perusteella.

Käyttöliittymäkomponentti muuntaa vastaanottamansa kaistanhallinta-asetukset sisäiseksi tietorakenteeksi, jota käytetään käyttöliittymän näkymän muodostamiseen, sekä asetusten muuttamisen mahdollistamiseen. Käyttöliittymä muuntaa tietorakenteen XML-muotoon, kun asetukset tulee lähettää kontrollikomponentille käyttöön otettavaksi kohdejärjestelmässä.

3.2 DTD-kuvaus

Kaistanhallinta-asetukset määritellään seuraavalla XML Document Type Definition (DTD) -kuvauksella:

```
<!ELEMENT system (interface*)>
<!ELEMENT interface (pfifofastqdisc | pfifoqdisc | bfifoqdisc |
                     redqdisc | htbqdisc)>
<!ATTLIST system
        hostname        CDATA #REQUIRED
        domain          CDATA #IMPLIED
        gatewayip       CDATA #IMPLIED
        gatewaydevice   CDATA #IMPLIED>
<!ATTLIST interface
        name            CDATA #REQUIRED
        ip              CDATA #IMPLIED
        netmask         CDATA #IMPLIED
        broadcast       CDATA #IMPLIED
        mac             CDATA #IMPLIED>
<!ELEMENT pfifofastqdisc EMPTY>
<!ELEMENT pfifoqdisc EMPTY>
<!ELEMENT bfifoqdisc EMPTY>
<!ELEMENT redqdisc EMPTY>
<!ATTLIST pfifoqdisc
        handle          CDATA #IMPLIED
        estimator       CDATA #IMPLIED
        limit           CDATA #IMPLIED>
<!ATTLIST bfifoqdisc
        handle          CDATA #IMPLIED
        estimator       CDATA #IMPLIED
        limit           CDATA #IMPLIED>
<!ATTLIST redqdisc
        handle          CDATA #IMPLIED
        estimator       CDATA #IMPLIED
        limit           CDATA #REQUIRED
        min             CDATA #REQUIRED
        max             CDATA #REQUIRED
        avpkt           CDATA #REQUIRED
        burst           CDATA #REQUIRED
        probability     CDATA #REQUIRED
        bandwidth       CDATA #REQUIRED
        ecn (true | false) #IMPLIED>
<!ELEMENT htbqdisc (htbclass+, u32filter*)>
<!ATTLIST htbqdisc
        handle          CDATA #IMPLIED
        estimator       CDATA #IMPLIED
        default         CDATA #IMPLIED
```

```
        r2q             CDATA #IMPLIED>
<!ELEMENT htbclass ((htbclass* | pfifoqdisc | bfifoqdisc |
                     redqdisc | htbqdisc)?, u32filter*)>
<!ATTLIST htbclass
        classid         CDATA #REQUIRED
        rate            CDATA #REQUIRED
        burst           CDATA #IMPLIED
        ceil            CDATA #IMPLIED
        cburst          CDATA #IMPLIED
        prio            CDATA #IMPLIED
        quantum         CDATA #IMPLIED>

<!ELEMENT u32filter (match*, police?)>
<!ELEMENT match EMPTY>
<!ELEMENT police EMPTY>
<!ATTLIST u32filter
        protocol        CDATA #IMPLIED
        classid         CDATA #IMPLIED
        police          CDATA #IMPLIED
        offset          CDATA #IMPLIED
        ht              CDATA #IMPLIED
        hashkey         CDATA #IMPLIED>
<!ATTLIST match
        sampletype (ip | ip6 | udp | tcp | icmp | u8 |
                    u16 | u32) #REQUIRED
        sampleargs      CDATA #REQUIRED>
<!ATTLIST police
        index           CDATA #IMPLIED
        buffer          CDATA #IMPLIED
        burst           CDATA #IMPLIED
        maxburst        CDATA #IMPLIED
        mtu             CDATA #IMPLIED
        avrate          CDATA #IMPLIED
        rate            CDATA #IMPLIED
        peakrate        CDATA #IMPLIED
        action (continue | drop | shot | pass | reclassify) #REQUIRED>
```

Kaikkien nopeutta ilmaisevien attribuuttien, kuten rate, arvot ilmaistaan lukuarvoina jotka tulkitaan ohjelmistossa bitteinä sekunnissa. Kaikkien määrää ilmaisevien attribuuttien, kuten limit, arvot ilmaistaan lukuarvoina jotka tulkitaan ohjelmistossa tavuiksi.

system on XML-dokumentin juurielementti. Elementin attribuutit sisältävät tietoa kohdejärjestelmästä. Tässä ohjelmiston versiossa hallintajärjestelmä ei voi muokata näitä tietoja. Elementin attribuutit ovat taulukossa 1.

interface kuvaa yhtä kohdejärjestelmän verkkoliityntää. Koska verkkoliityntöjä voi yhdessä järjestelmässä olla useita, saa elementti toistua yhdessä XML-dokumentissa mielivaltaisen monta kertaa. Elementin attribuutit ovat taulukossa 2.

Nimi	*Tyyppi*	*Tietosisältö*
hostname	merkkijono	Kohdejärjestelmän verkkonimi.
domain	merkkijono	Kohdejärjestelmän toimialue.
gatewayip	merkkijono	Kohdejärjestelmän IPv4-liikenteen oletusreitittimen IP-osoite.
gatewaydev	merkkijono	Kohdejärjestelmän oletusreitittimen verkkoliityntä.

Taulukko 1: system-elementin attribuutit.

Nimi	*Tyyppi*	*Tietosisältö*
name	merkkijono	Kohdejärjestelmän verkkoliitynnästä käyttämä nimi, esimerkiksi ”eth0”.
ip	merkkijono	Verkkoliitynnän IP-osoite.
netmask	merkkijono	Verkkoliitynnän IP-verkon aliverkon peite, esimerkiksi ”255.255.255.0”.
broadcast	merkkijono	Verkkoliitynnän IP-broadcast-osoite.
mac	merkkijono	Verkkoliitynnän MAC-osoite, esimerkiksi ”00:50:56:C0:00:10”.

Taulukko 2: interface-elementin attribuutit.

pfifofastqdisc-elementti kuvaa pfifo_fast-jonomallia. Elementti voi esiintyä vain interface-elementin lapsena. Elementti esiintyy oletusarvoisesti liitettynä jokaiseen Linux-järjestelmän verkkoliityntään, eikä sillä ole käyttäjän muokattavissa olevia attribuutteja.

pfifoqdisc-elementti kuvaa pfifo-jonomallia. Elementti voi esiintyä joko interface-elementin tai htbclass-elementin lapsena. Elemetin attribuutit ovat taulukossa 3.

Nimi	*Tyyppi*	*Tietosisältö*
handle	merkkijono	Yksilöllinen tunniste, esimerkiksi ”10:”.
estimator	merkkijono	Attribuutin merkitystä ei tiedetä.
limit	kokonaisluku	Pakettijonon pituus, esimerkiksi ”30”.

Taulukko 3: pfifoqdisc-elementin attribuutit.

bfifoqdisc-elementti kuvaa bfifo-jonomallia. Elementti voi esiintyä joko interface-elementin tai htbclass-elementin lapsena. Elementin attribuutit ovat taulukossa 4.

redqdisc-elementti kuvaa RED-jonomallia. Elementti voi esiintyä joko interface-elementin tai htbclass-elementin lapsena. Elementin attribuutit ovat taulu-

Nimi	*Tyyppi*	*Tietosisältö*
handle	merkkijono	Yksilöllinen tunniste, esimerkiksi "20:".
estimator	merkkijono	Attribuutin merkitystä ei tiedetä.
limit	kokonaisluku	Jonon pituus tavuina, esimerkiksi "30000".

Taulukko 4: bfifoqdisc-elementin attribuutit.

kossa 5.

Nimi	*Tyyppi*	*Tietosisältö*
handle	merkkijono	Yksilöllinen tunniste, esimerkiksi "30:".
estimator	merkkijono	Attribuutin merkitystä ei tiedetä.
limit	kokonaisluku	Jonon pituus tavuina, esimerkiksi "30000".
min	kokonaisluku	Keskimääräinen jonon pituus, jolloin RED saattaa pudottaa paketteja.
max	kokonaisluku	Jonon keskimääräinen pituus, jolloin pakettien pudotuksen todennäköisuus on suurimmillaan.
avpkt	kokonaisluku	Keskimääräinen paketin koko.
burst	kokonaisluku	Kuinka nopeasti todellinen koko vaikuttaa jonon keskimääräiseen kokoon.
probability	kokonaisluku	Pakettien suurin hylkäämistodennäköisyys.
bandwidth	kokonaisluku	Verkkoliitynnän nopeus.
ecn	totuusarvo	Käytetäänkö IP-kehyksen ECN-kenttää [RFB01].

Taulukko 5: redqdisc-elementin attribuutit.

htbqdisc-elementti kuvaa luokallista HTB-jonomallia. Elementti voi esiintyä joko interface-elementin tai htbclass-elementin lapsena. Elementin attribuutit ovat taulukossa 6.

Nimi	*Tyyppi*	*Tietosisältö*
handle	merkkijono	Yksilöllinen tunniste, esimerkiksi "1:".
estimator	merkkijono	Attribuutin merkitystä ei tiedetä.
default	kokonaisluku	Oletusluokka, johon paketit ohjataan. Esimerkiksi "20".
r2q	kokonaisluku	Attribuutin merkitystä ei tiedetä.

Taulukko 6: htbqdisc-elementin attribuutit.

htbclass-elementti kuvaa HTB-jonomallin luokkaa. Htbclass-elementti voi esiintyä joko htbqdisc-elementin tai toisen htbclass-elementin lapsena. Elementin attribuutit ovat taulukossa 7.

Nimi	*Tyyppi*	*Tietosisältö*
classid	merkkijono	Yksilöllinen tunniste, esimerkiksi ”1:20”.
rate	merkkijono	Luokan kaistanleveys.
burst	kokonaisluku	Tavumäärä, jonka luokka voi lähettää paketteja ceil-nopeudella, jos se ylittää kaistanleveytensä.
ceil	kokonaisluku	Maksiminopeus jolla luokka voi lähettää paketteja, jos sen vanhempi alittaa mahdollisen suurimman kaistankäyttönsä.
cburst	kokonaisluku	Vastaava burst-määrä ceil-arvolle.
prio	kokonaisluku	Tämän luokan prioriteetti. Pienimmät lähettävät ensin.
quantum	kokonaisluku	Kuinka monta tavua voidaan lähettää kerrallaan, kunnes seuraava luokka saa vuoron.

Taulukko 7: htbclass-elementin attribuutit.

u32filter-elementti kuvaa u32-suodatinta. Elementti voi esiintyä joko htbqdisc- tai htbclass-elementin lapsena. Elementin attribuutit ovat taulukossa 8.

Nimi	*Tyyppi*	*Tietosisältö*
protocol	merkkijono	Käytetty protokolla, esimerkiksi ”ip”.
classid	merkkijono	Mihin luokkaan tämä suodatin ohjaa pakettiliikenteen, esimerkiksi ”1:20”.
offset	merkkijono	Attribuutin merkitystä ei tiedetä.
ht	merkkijono	Attribuutin merkitystä ei tiedetä.
hashkey	merkkijono	Attribuutin merkitystä ei tiedetä.

Taulukko 8: u32filter-elementin attribuutit.

match-elementti kuvaa u32-suodattimen match-parametria. Elementti voi esiintyä vain u32filter-elementin lapsena, ja se voi toistua mielivaltaisen monta kertaa. Elementin attribuutit ovat taulukossa 9.

police-elementti kuvaa u32-suodattimen police-parametria. Elementti voi esiintyä vain u32filter-elementin lapsena, ja se voi toistua korkeintaan yhden kerran. Elementin attribuutit ovat taulukossa 10.

Nimi	*Tyyppi*	*Tietosisältö*
sampletype	vaihtoehdot	Match-parametrin rajoitin. Attribuutti voi saada vahtoehdot ip, ip6, udp, tcp, icmp, u32, u16 tai u8.
sampleargs	merkkijono	Match-parametrin arvo.

Taulukko 9: match-elementin attribuutit.

Nimi	*Tyyppi*	*Tietosisältö*
index	kokonaisluku	Attribuutin merkitystä ei tiedetä.
burst	kokonaisluku	Suodattimen sisäisen jonon purskenopeus.
maxburst	kokonaisluku	Suodattimen sisäisen jonon maksimipurske.
mtu	kokonaisluku	Suodattimen suurimman mahdollisen siirtoyksikön koko.
avrate	kokonaisluku	Suodattimen sisäisen jonon keskimääräinen nopeus.
rate	kokonaisluku	Suodattimen sisäisen jonon nopeus.
peakrate	kokonaisluku	Suodattimen sisäisen jonon maksiminopeus.
action	vaihtoehdot	Attribuutti voi saada vaihtoehdot continue, drop, pass, reclassify.

Taulukko 10: police-elementin attribuutit.

4 STCCP-protokolla

Simple Traffic Control Configuration Protocol (STCCP) on Paketti-ryhmän määrittelemä sovellustason protokolla kaistanhallinta-asetusten ja statistiikan siirtoon. Protokolla perustuu asiakas-palvelin -mallin mukaiseen protokolla-arkkitehtuuriin ja hyödyntää TCP/IP-protokollaa. Kaikki protokollan mukaiset palvelupyynnöt sekä vastaukset sijaitsevat protokollan XML-muotoisessa viestikehyksessä.

4.1 Toimintamalli

STTCP-protokollan toimintamalli on seuraava:

- Asiakas avaa TCP-yhteyden palvelimeen.
- Asiakas lähettää palvelimelle pyynnön.
- Palvelin vastaa asiakkaan pyyntöön.
- Saatuaan vastauksen palvelimelta asiakas voi lähettää uuden pyynnön.

- Asiakas katkaisee yhteyden palvelimeen lähettämällä QUIT-pyynnön ja sulkemalla TCP-yhteyden.

Seuraavalla kaavio havainnollistaa protokollan käyttöä. Merkintä * tarkoittaa, että tapahtumaa voidaan toistaa.

```
      Asiakas ---- TCP/IP-yhteyden avaus ---> Palvelin

.---> Asiakas ------------ pyyntö ----------> Palvelin
|     Asiakas <----------- vastaus ---------- Palvelin
|_____|
   *
      Asiakas --------- QUIT-pyyntö --------> Palvelin
      Asiakas --- TCP/IP-yhteyden lopetus --> Palvelin
```

4.2 STCCP-viestin rakenne

STCCP-viestin rakenteen kuvauksessa käytetään seuraavia merkintöjä:

```
          LF = rivinvaihto
               heksadesimaalilukuna 0xA, desimaalilukuna 10

kokonaisluku = ei-negatiivinen kokonaisluku k
               väliltä 0 <= k < 2^32 = 4294967296

      merkki = mielivaltainen UTF-8 -merkistön merkki

  merkkijono = mielivaltainen jono xml-syntaksin hyväksymiä
               merkkejä
```

STCCP-viestit käyttävät Unicode UTF-8 -merkistöä. Viestien muoto on:

```
versionumero LF
pituus LF
xml-kehys
```

STCCP-protokollan versionumero ilmoitetaan muodossa:

```
Protocol:STCCP/kokonaisluku
```

Protokollan ensimmäisessä versiossa kokonaisluvun arvo on 1, toisessa 2, jne. Kentässä ei saa olla muita kuin yllä määriteltyjä merkkejä. Pituus-kentän muoto on:

```
Length:kokonaisluku
```

Kokonaisluku ilmoittaa viimeisessä kentässä olevan XML-kehyksen pituuden tavuina. Kentässä ei saa olla muita kuin yllä määriteltyjä merkkejä. Viimeisessä kentässä on XML-viestikehys, jonka rakenne selitetään luvussa 4.3.

Esimerkki GET_STATISTICS_TC -pyynnöstä:

```
Protocol:STCCP/1 LF
Length:54 LF
<MESSAGE><TYPE>GET_STATISTICS_TC</TYPE><DATA /></MESSAGE>
```

4.3 XML-kehyksen rakenne

STCCP-viestin sisältö tallennetaan XML-muotoiseen viestikehykseen. XML-kehyksen DTD-kuvaus on seuraava:

```
<!ELEMENT MESSAGE ((SEQUENCE, ACK)?, TYPE, DATA)>
<!ELEMENT SEQUENCE(#CDATA)>
<!ELEMENT ACK(#CDATA)>
<!ELEMENT TYPE(#CDATA)>
<!ELEMENT DATA(#CDATA)>
```

Valinnainen SEQUENCE, ACK -pari on sisällytetty DTD-kuvaukseen tulevaisuuden laajennuksia varten. Ne eivät ole tässä protokollan versiossa käytössä. TYPE- ja DATA-elementit ovat pakolliset.

TYPE-elementin sisältö kertoo viestin tyypin. Viestityyppejä on kahdenlaisia: pyyntöjä ja vastauksia. Pyyntöjä lähettää vain asiakas ja vastauksia vain palvelin. Palvelin vastaa jokaiseen asiakkaan lähettämään pyyntöön QUIT-pyyntöä lukuunottamatta. Pyynnöt ja vastaukset kirjoitetaan aina isoilla kirjaimilla.

DATA-elementin sisältö on siirrettävä tieto, joita ovat esimerkiksi kaistanhallinta-asetukset, virheilmoitusten sisältö sekä statistiikkatiedot.

4.3.1 Pyynnöt

Jokaisesta pyynnöstä kerrotaan tässä viestityyppi, palvelimen lähettämä vastaus sekä pyynnön käsittelyssä mahdollisia virhetilanteita. Hyväksyttyyn pyyntöön palvelin vastaa REPLY-vastauksella, virheelliseen ERROR-vastauksella. Kaikille pyynnöille yhteiset virhetilanteet löytyvät ERROR-vastauksen kuvauksesta.

GET_SETTINGS_XML

Kuvaus: GET_SETTINGS_XML -pyynnöllä asiakas pyytää nykyisiä kaistanhallinta-asetuksia palvelimelta XML-muodossa. Palvelin ei käsittele viestin DATA-elementtiä.

Vastaus: DATA-elementti sisältää kaistanhallinta-asetukset XML-muodossa.

Virhetilanteet: kts. ERROR

```
<MESSAGE><TYPE>GET_SETTINGS_XML</TYPE><DATA /></MESSAGE>
```

GET_SETTINGS_TC

Kuvaus: GET_SETTINGS_TC -pyynnöllä asiakas pyytää nykyisiä asetuksia palvelimelta tc-komentomuodossa. Palvelin ei käsittele viestin DATA-elementtiä.

Vastaus: DATA-elemetti sisältää asetukset tc-komentoina tekstimuodossa. Tc-komennot erotetaan toisistaan rivinvaihtomerkillä.

Virhetilanteet: kts. ERROR

```
<MESSAGE><TYPE>GET_SETTINGS_TC</TYPE><DATA /></MESSAGE>
```

GET_STATISTICS_XML

Kuvaus: GET_STATISTICS_XML -pyynnöllä asiakas pyytää kaistanhallinta-asetusten statistiikkaa palvelimelta XML-muodossa. Palvelin ei käsittele viestin DATA-elementtiä.

Vastaus: DATA-elementti sisältää statistiikkatiedot XML-muodossa. XML-muotoa ei ole määritelty tässä protokollan versiossa.

Virhetilanteet: kts. ERROR

```
<MESSAGE><TYPE>GET_STATISTICS_XML</TYPE><DATA /></MESSAGE>
```

GET_STATISTICS_TC

Kuvaus: GET_STATISTICS_TC -pyynnöllä asiakas pyytää kaistanhallinta-asetusten statistiikkaa palvelimelta tc-työkalun tulostamassa muodossa. Palvelin ei käsittele viestin DATA-elementtiä.

Vastaus: DATA-elementti sisältää statistiikkatiedot tc-komennon käyttämässä muodossa.

Virhetilanteet: kts. ERROR

```
<MESSAGE><TYPE>GET_STATISTICS_TC</TYPE><DATA /></MESSAGE>
```

COMMIT

Kuvaus: COMMIT -pyynnöllä asiakas pyytää palvelinta ottamaan käyttöön DATA-elementin XML-muotoiset kaistanhallinta-asetukset.

Vastaus: DATA-elementti sisältää käyttöön otetut asetukset XML-muodossa.

Virhetilanteet: DATA-elementti on tyhjä.
DATA-elementin XML-muoto ei ole kelvollinen.
DATA-elementin kuvaamat kaistanhallinta-asetukset eivät ole kelvolliset.

```
<MESSAGE><TYPE>COMMIT</TYPE><DATA>merkkijono</DATA></MESSAGE>
```

TRANSFORM_TC_TO_XML

Kuvaus: TRANSFORM_TC_TO_XML -pyynnöllä asiakas pyytää palvelinta muuntamaan DATA-elementin sisältämät tc-komentomuotoiset kaistanhallinta-asetukset XML-muotoon.

Vastaus: DATA-elementti sisältää kaistanhallinta-asetukset XML-muodossa.

Virhetilanteet: DATA-elementti on tyhjä.
Teksti ei muodostu tc-komennoista.
Tc-komennot ovat kelvottomia.
Tc-komennot eivät muodosta kelvollisia kaistanhallinta-asetuksia.

```
<MESSAGE><TYPE>TRANSFORM_TC_TO_XML</TYPE><DATA>merkkijono</DATA></MESSAGE>
```

TRANSFORM_XML_TO_TC

Kuvaus: TRANSFORM_XML_TO_TC -pyynnöllä asiakas pyytää palvelinta muuttamaan DATA-elementin sisältämät asetukset tc-komennoiksi. Asetusten tulee olla XML-muodossa.

Vastaus: DATA-elementti sisältää kaistanhallinta-asetukset tc-komentoina.

Virhetilanteet: DATA-elementti on tyhjä.
DATA-elementti ei muodostu XML-viestistä.
DATA-elementin XML-muoto ei ole kelvollista.

```
<MESSAGE><TYPE>TRANSFORM_XML_TO_TC</TYPE><DATA>merkkijono</DATA></MESSAGE>
```

QUIT

Kuvaus: QUIT- pyynnöllä asiakas katkaisee yhteyden palvelimeen. Palvelin ei käsittele viestin DATA-elementtiä.

Vastaus: Palvelin ei lähetä vastausta.

Virhetilanteet: -

```
<MESSAGE><TYPE>QUIT</TYPE><DATA /></MESSAGE>
```

4.3.2 Vastaukset

REPLY

Kuvaus: Palvelin vastaa REPLY-vastauksella asiakkaan
GET_SETTINGS_TC-,
GET_SETTING_XML-,
GET_STATISTICS_TC-,
GET_STATISTICS_XML-,
COMMIT-,
TRANSFORM_XML_TO_TC- ja
TRANSFORM_TC_TO_XML-pyyntöihin.
Vastauksen DATA-elementti sisältää pyydetyn tiedon.

```
<MESSAGE><TYPE>REPLY</TYPE><DATA>merkkijono</DATA></MESSAGE>
```

ERROR

Kuvaus: Palvelin lähettää ERROR-vastauksen, mikäli asiakkaan tekemän pyynnön suorituksessa ilmenee virhe. Vastauksen DATA-elementin sisältönä on lisätietoa virheestä. Mahdollisia virhetilanteita ovat:

Virheellisesti muotoiltu viesti.
Viestissä ei tuettuja kaistanhallinta-asetuksia.
Virhe kaistanhallinta-asetuksia muunnettaessa.
Virhe kaistanhallinta-asetusten käyttöönotossa.
Virhe pyydettäessä kaistanhallinta-asetuksia järjestelmältä.
Virhe pyydettäessä statistiikkaa järjestelmältä.

```
<MESSAGE><TYPE>ERROR</TYPE><DATA>merkkijono</DATA></MESSAGE>
```

4.4 Esimerkkikeskustelu

Luvussa näytetään kaksi esimerkkiä protokollan mukaisesta keskustelusta asiakkaan ja palvelimen välillä. Ensimmäisessä esimerkissä asiakas pyytää statistiikkaa palvelimelta. Toisessa esimerkissä asiakas pyytää palvelinta muuttamaan antamansa tc-komennot XML-muotoon.

4.4.1 Esimerkki GET_STATISTICS_TC-pyynnöstä

Asiakas:

```
Protocol:STCCP/1 LF
Length:54 LF
<MESSAGE><TYPE>GET_STATISTICS_TC</TYPE><DATA /></MESSAGE>
```

Palvelin:

```
Protocol:STCCP/1 LF
Length:1976 LF
<MESSAGE><TYPE>REPLY</TYPE><DATA>[qdisc]
qdisc sfq 13: quantum 1492b perturb 10sec
 Sent 1569980 bytes 11521 pkts (dropped 0, overlimits 0)

 qdisc sfq 12: quantum 1492b perturb 10sec
 Sent 0 bytes 0 pkts (dropped 0, overlimits 0)

 qdisc sfq 11: quantum 1492b perturb 10sec
 Sent 0 bytes 0 pkts (dropped 0, overlimits 0)

 qdisc sfq 10: quantum 1492b perturb 10sec
 Sent 0 bytes 0 pkts (dropped 0, overlimits 0)

 qdisc htb 1: r2q 1 default 13 direct_packets_stat 0
 Sent 1569980 bytes 11521 pkts (dropped 0, overlimits 453)

[class]
class htb 1:11 parent 1:1 leaf 11: prio 1 rate 30Kbit ceil 126Kbit
burst 1637b cburst 1760b
 Sent 0 bytes 0 pkts (dropped 0, overlimits 0)
 lended: 0 borrowed: 0 giants: 0
 tokens: 349439 ctokens: 89446

class htb 1:1 root rate 126Kbit ceil 126Kbit burst 1760b cburst 1760b
 Sent 1569888 bytes 11520 pkts (dropped 0, overlimits 0)
 rate 1bps
 lended: 2973 borrowed: 0 giants: 0
 tokens: 84977 ctokens: 84977

class htb 1:10 parent 1:1 leaf 10: prio 0 rate 10Kbit ceil 126Kbit
burst 1611b cburst 1760b
 Sent 0 bytes 0 pkts (dropped 0, overlimits 0)
 lended: 0 borrowed: 0 giants: 0
 tokens: 1031680 ctokens: 89446

class htb 1:13 parent 1:1 leaf 13: prio 3 rate 10Kbit ceil 126Kbit
burst 1611b cburst 1760b
 Sent 1569888 bytes 11520 pkts (dropped 0, overlimits 0)
 rate 3bps
 lended: 8547 borrowed: 2973 giants: 0
 tokens: 975360 ctokens: 84977

class htb 1:12 parent 1:1 leaf 12: prio 2 rate 75Kbit ceil 126Kbit
burst 1695b cburst 1760b
 Sent 0 bytes 0 pkts (dropped 0, overlimits 0)
 lended: 0 borrowed: 0 giants: 0
 tokens: 144724 ctokens: 89446
```

```
[filter]
filter parent 1: protocol ip pref 1 fw
filter parent 1: protocol ip pref 1 fw handle 0xb classid 1:11
filter parent 1: protocol ip pref 2 fw
filter parent 1: protocol ip pref 2 fw handle 0xc classid 1:12
filter parent 1: protocol ip pref 3 fw
filter parent 1: protocol ip pref 3 fw handle 0xd classid 1:13
filter parent 1: protocol ip pref 49152 fw
filter parent 1: protocol ip pref 49152 fw handle 0xa classid 1:10
</DATA></MESSAGE>
```

Asiakas:

```
Protocol:STCCP/1 LF
Length:44 LF
<MESSAGE><TYPE>QUIT</TYPE><DATA /></MESSAGE>
```

4.4.2 Laajempi esimerkki

Asiakas:

```
Protocol:STCCP/1 LF
Length:384 LF
<MESSAGE><TYPE>TRANSFORM_TC_TO_XML</TYPE><DATA>
tc qdisc add dev eth0 root handle 1: htb default 10 r2q 9
tc class add dev eth0 parent 1: classid 1:10 htb rate 2mbit burst 3000
tc qdisc add dev eth0 handle 10: parent 1:10 pfifo limit 100
tc qdisc add dev eth1 root handle 1: htb
tc class add dev eth1 parent 1: classid 1:10 htb rate 2mbit burst 3000
tc qdisc add dev eth1 parent 1:10 pfifo
tc qdisc add dev eth2 root pfifo
</DATA></MESSAGE>
```

Palvelin:

```
Protocol:STCCP/1 LF
Length:901 LF
<MESSAGE><TYPE>REPLY</TYPE><DATA>
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE system SYSTEM "kaistanhallinta-asetukset.dtd">
<system hostname="testikone" domain="testi.net" gatewayip="192.168.0.65"
        gatewaydevice="eth0">
    <interface name="eth0" ip="192.168.0.66" netmask="255.255.255.224"
               broadcast="192.168.0.95" mac="01:23:45:67:89:0A">
        <htbqdisc handle="1:" default="10" r2q="9">
            <htbclass classid="1:10" rate="2mbit" burst="3000">
                <pfifoqdisc handle="10:" limit="100"/>
            </htbclass>
        </htbqdisc>
    </interface>
```

```
    <interface name="eth1" ip="192.168.0.67" netmask="255.255.255.224"
               broadcast="192.168.0.95" mac="BC:DE:F0:12:34:56">
        <htbqdisc handle="1:">
            <htbclass classid="1:10" rate="2mbit" >
                <pfifoqdisc/>
            </htbclass>
        </htbqdisc>
    </interface>
    <interface name="eth2" ip="192.168.0.68" netmask="255.255.255.224"
               broadcast="192.168.0.95" mac="78:9A:BC:DE:F0:12">
        <pfifoqdisc/>
    </interface>
</system>
</DATA></MESSAGE>
```

Asiakas:

```
Protocol:STCCP/1 LF
Length:899 LF
<MESSAGE><TYPE>COMMIT</TYPE><DATA>
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE system SYSTEM "kaistanhallinta-asetukset.dtd">
<system hostname="testikone" domain="testi.net" gatewayip="192.168.0.65"
        gatewaydevice="eth0">
    <interface name="eth0" ip="192.168.0.66" netmask="255.255.255.224"
               broadcast="192.168.0.95" mac="01:23:45:67:89:0A">
        <htbqdisc handle="1:" default="10" r2q="9">
            <htbclass classid="1:10" rate="2mbit" burst="3000">
                <pfifoqdisc handle="10:" limit="100"/>
            </htbclass>
        </htbqdisc>
    </interface>
    <interface name="eth1" ip="192.168.0.67" netmask="255.255.255.224"
               broadcast="192.168.0.95" mac="BC:DE:F0:12:34:56">
        <htbqdisc handle="1:">
            <htbclass classid="1:10" rate="2mbit" >
                <pfifoqdisc/>
            </htbclass>
        </htbqdisc>
    </interface>
    <interface name="eth2" ip="192.168.0.68" netmask="255.255.255.224"
               broadcast="192.168.0.95" mac="78:9A:BC:DE:F0:12">
        <pfifoqdisc/>
    </interface>
</system>
</DATA></MESSAGE>
```

Palvelin:

```
Protocol:STCCP/1 LF
Length:901 LF
```

```
<MESSAGE><TYPE>REPLY</TYPE><DATA>
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE system SYSTEM "kaistanhallinta-asetukset.dtd">
<system hostname="testikone" domain="testi.net" gatewayip="192.168.0.65"
        gatewaydevice="eth0">
    <interface name="eth0" ip="192.168.0.66" netmask="255.255.255.224"
               broadcast="192.168.0.95" mac="01:23:45:67:89:0A">
        <htbqdisc handle="1:" default="10" r2q="9">
            <htbclass classid="1:10" rate="2mbit" burst="3000">
                <pfifoqdisc handle="10:" limit="100"/>
            </htbclass>
        </htbqdisc>
    </interface>
    <interface name="eth1" ip="192.168.0.67" netmask="255.255.255.224"
               broadcast="192.168.0.95" mac="BC:DE:F0:12:34:56">
        <htbqdisc handle="1:">
            <htbclass classid="1:10" rate="2mbit" >
                <pfifoqdisc/>
            </htbclass>
        </htbqdisc>
    </interface>
    <interface name="eth2" ip="192.168.0.68" netmask="255.255.255.224"
               broadcast="192.168.0.95" mac="78:9A:BC:DE:F0:12">
        <pfifoqdisc/>
    </interface>
</system>
</DATA></MESSAGE>
```

Asiakas:

```
Protocol:STCCP/1 LF
Length:44 LF
<MESSAGE><TYPE>QUIT</TYPE><DATA /></MESSAGE>
```

5 Käyttöliittymäkomponentti

Käyttäjä on yhteydessä järjestelmään käyttöliittymän kautta, joka sijaitsee käyttöliittymäkomponentissa. Tämä luku esittelee käyttöliittymäkomponentin toiminnallisuuden.

5.1 Paketti-sovelluksen käynnistys

Käyttäjä käynnistää paketti-sovelluksen antamalla sille parametreinä kohdejärjestelmän osoitteen sekä porttin, johon sovellus ottaa yhteyden. Käyttäjälle tulostetaan tekstitilassa tietoa yhteyden muodostamisesta. Mikäli pyyntö ei onnistu vaan yhteysajastin laukeaa tai kontrollikomponentti vastaa

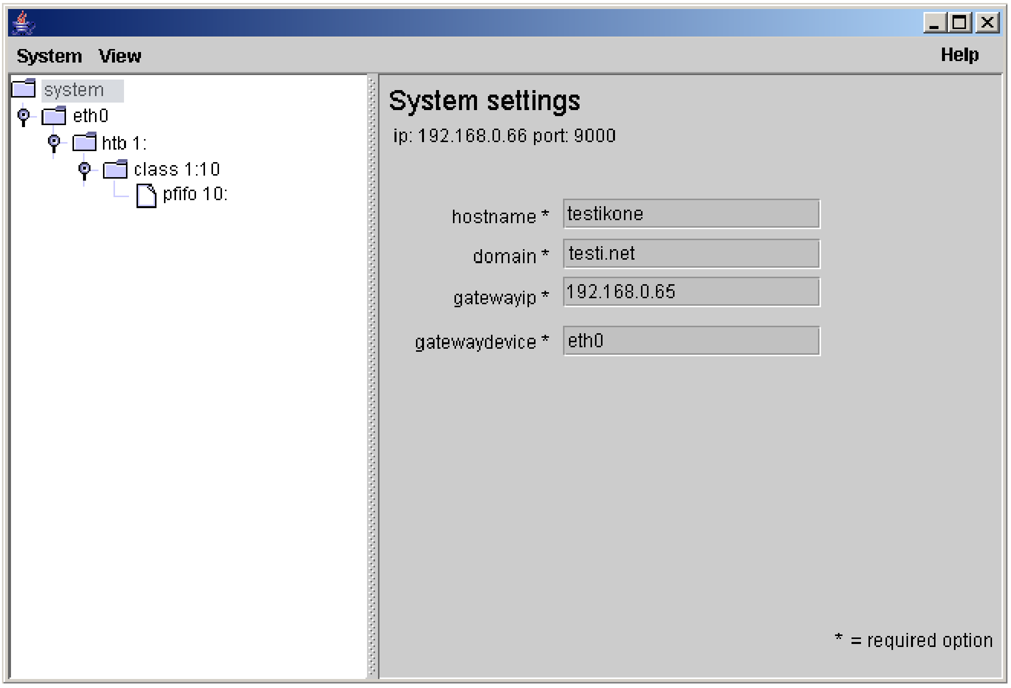
ERROR-viestillä, käyttäjälle tulostetaan virheilmoitus ja sovelluksen suoritus päättyy. Jos pyyntö onnistui, sovelluksen graafinen käyttöliittymä käynnistyy. Käynnistyksen jälkeen sovellus pyytää automaattisesti kaistanhallinta-asetukset kohdejärjestelmältä.

5.2 Käyttöliittymän ulkoasu

Seuraavassa esitellään käyttöliittymän toiminta käyttäjän näkökulmasta.

5.2.1 Käyttöliittymän perusnäkymä

Graafinen käyttöliittymä käynnistyy perusnäkymään (kuva 3). Käynnistyksen yhteydessä kohdejärjestelmältä pyydetyt kaistanhallinta-asetukset esitetään *puunäkymänä* käyttöliittymäikkunan vasemmassa kehyksessä. Puunäkymän elementtien asetukset näkyvät käyttöliittymäikkunan oikeassa kehyksessä *asetusnäkymänä*.



Kuva 3: Käyttöliittymä: perusnäkymä

5.2.2 Puunäkymän muokkaus

Jokainen jonomalli, suodatin, luokka ja verkkoliittymä kuvataan elementteinä puunäkymässä. Puunäkymää voidaan muokata hiirellä. Hiiren oikealla painikkeella elementistä avautuu kontekstivalikko, jonka avulla elementti voidaan poistaa tai lisätä siihen alielementti. Elementtejä poistettaessa poistuvat myös elementin alielementit.

Taulukossa 11 on lueteltu puunäkymän elementit, sekä kunkin elementin kohdalla tieto, voiko elementin poistaa, sekä mitkä elementit on mahdollista sijoittaa kyseisen elementin alielementiksi. Puun hierarkia noudattaa luvussa 3 esitettyä kaistanhallinta-asetusten DTD-kuvausta.

Elementti	Voiko poistaa	Mahdolliset alielementit
system	ei	interface
interface	ei	pfifofastqdisc, pfifoqdisc, bfifoqdisc, redqdisc, htbqdisc
redqdisc	kyllä	-
pfifoqdisc	kyllä	-
bfifoqdisc	kyllä	-
police	kyllä	-
htbqdisc	kyllä	htbclass, u32filter-
pfifofastqdisc	kyllä	-
htbclass	kyllä	pfifoqdisc, bfifoqdisc, redqdisc, htbqdisc, u32filter
u32filter	kyllä	match, police
match	kyllä	-

Taulukko 11: Puunäkymän elementit

Lisäys

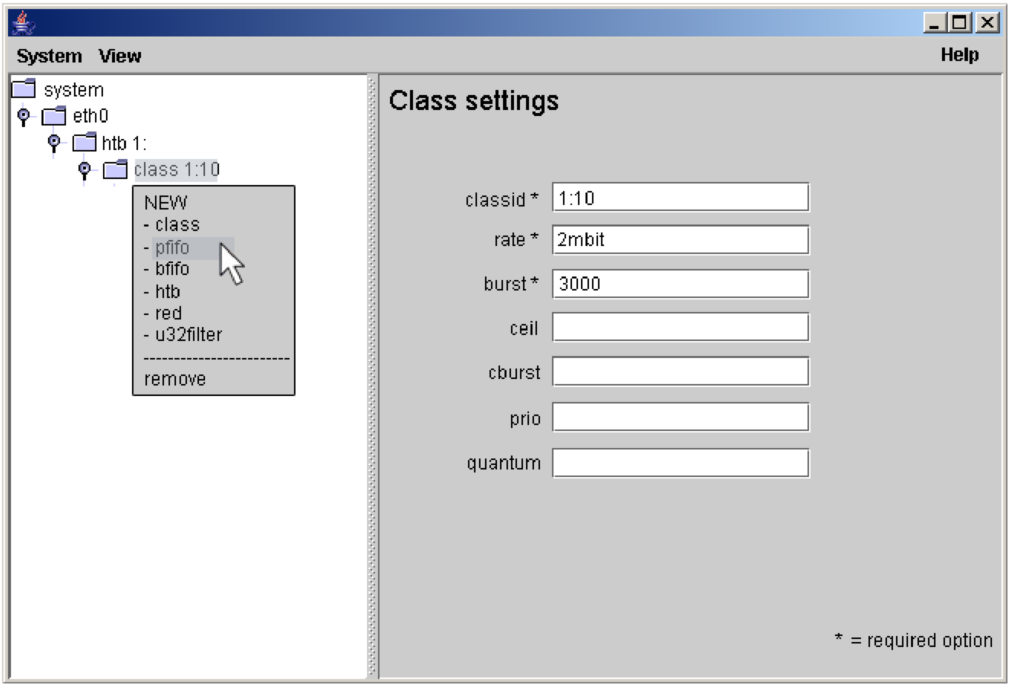
Uudet elementit lisätään puunäkymään olemassa olevan elementin alle. Elementistä avautuu hiiren oikealla painikkeella kontekstivalikko, josta valitaan lisättävä elementti (kuva 4). Valikossa näkyvät ainoastaan ne elementit, joita saa lisätä valitun elementin alle. Lisätty elementti ilmestyy puunäkymään välittömästi (kuva 5).

Poisto

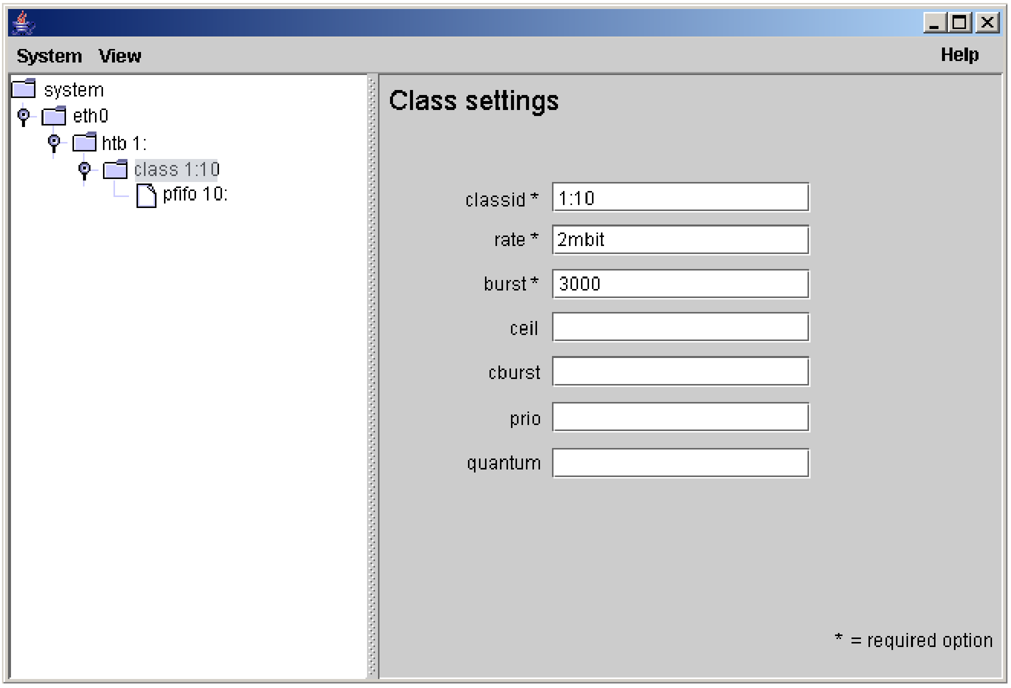
Elementti poistetaan puunäkymästä avaamalla elementin kontekstivalikko hiiren oikealla painikkeella ja valitsemalla valikosta ”remove” (kuva 6). Poistettu elementti alielementteineen häviää puunäkymästä välittömästi (kuva 7).

5.2.3 Elementin asetuksien muokkaus

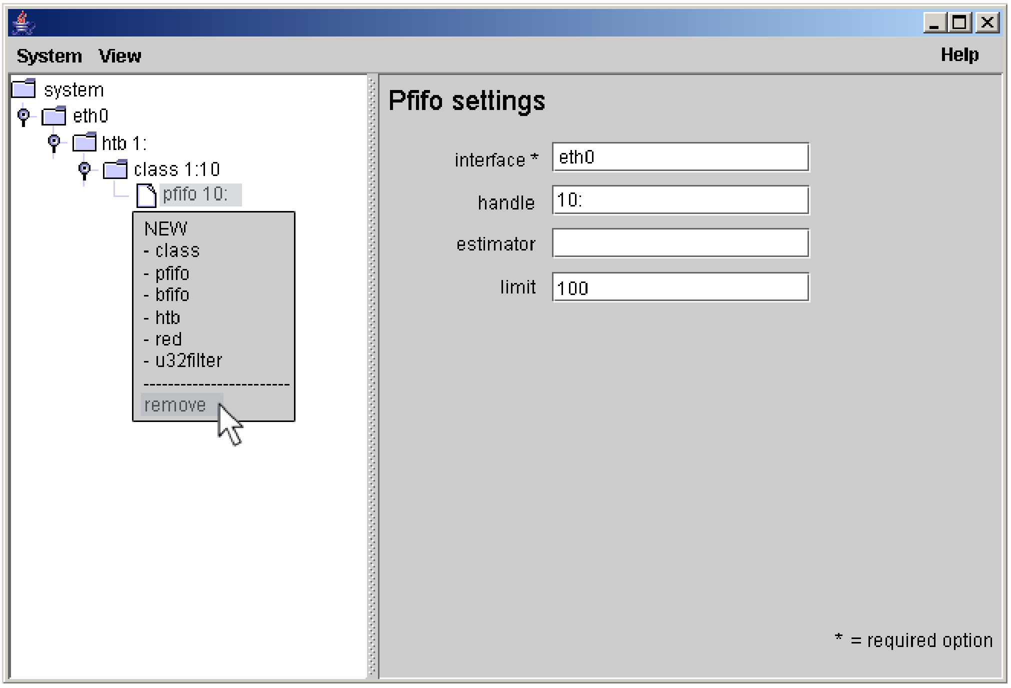
Elementin asetuksien muokkaaminen aloitetaan valitsemalla elementti puunäkymästä hiiren vasemmalla painikkeella. Valitun elementin asetukset ilmestyvät asetusnäkymään tekstikenttinä, jotka sisältävät valitun elementin



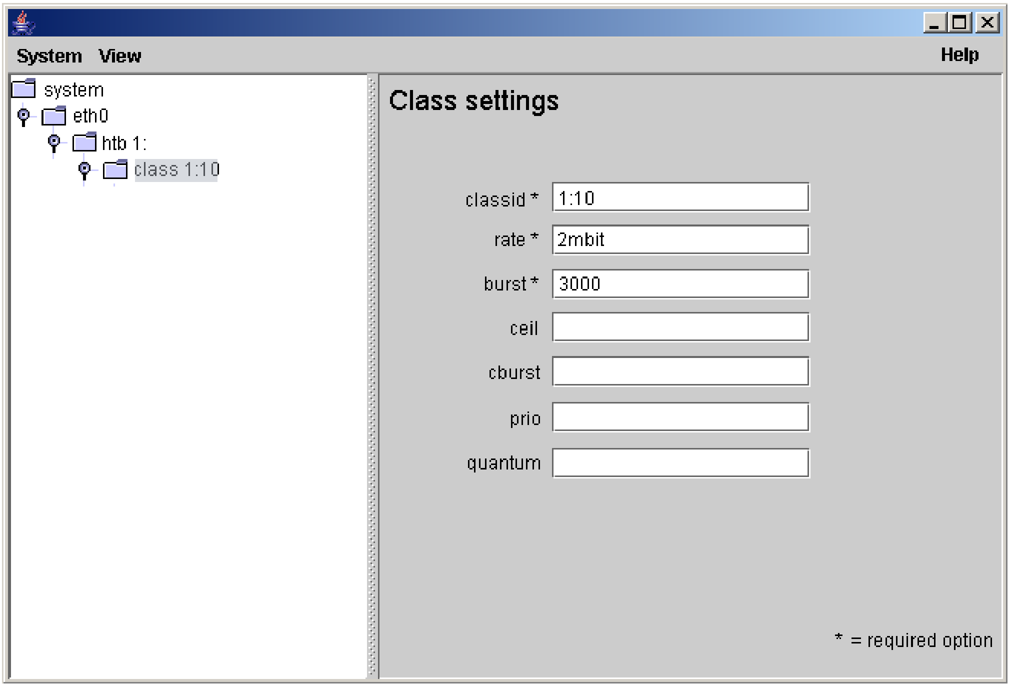
Kuva 4: Käyttöliittymä: elementin lisäys



Kuva 5: Käyttöliittymä: puunäkymä lisäyksen jälkeen.



Kuva 6: Käyttöliittymä: elementin poisto

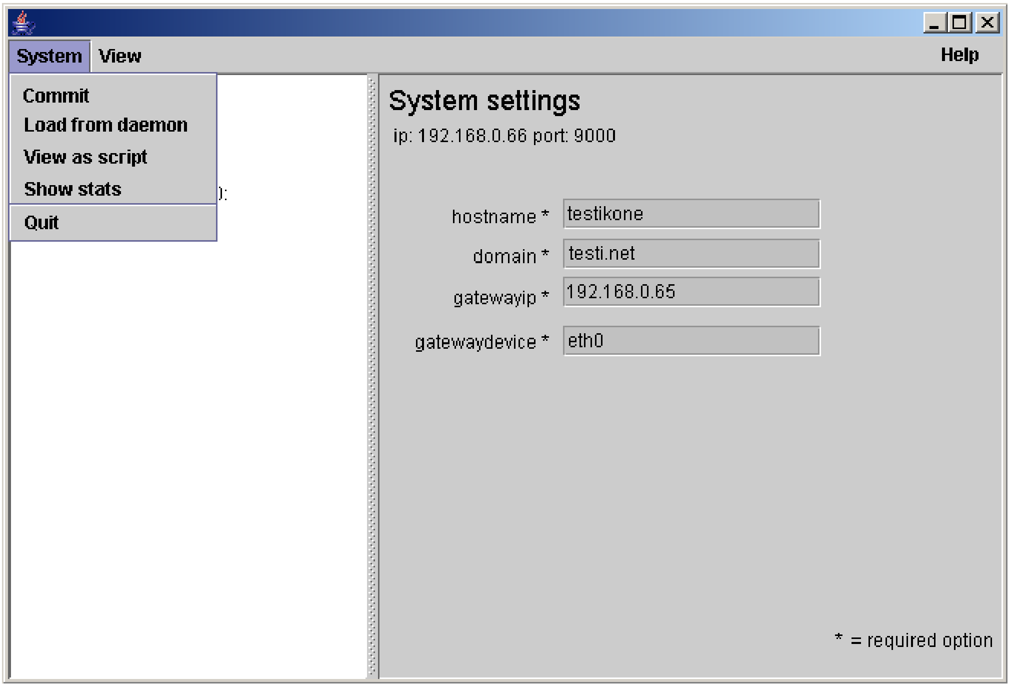


Kuva 7: Käyttöliittymä: puunäkymä poiston jälkeen.

asetukset. Useimmat asetukset ovat vapaasti muokattavissa. Tähdellä merkittyjen asetuksien määritys on pakollista. Asetukset, joiden muokkaaminen ei ole mahdollista, esitetään harmaalla pohjalla.

5.2.4 Käyttöliittymän valikot

Käyttöliittymässä on kolme valikkoa: System, View ja Help. System-valikko koostuu Commit, Load from daemon, View as script, Show stats ja Quit -toiminnoista (kuva 8). View-valikko sisältää Interfaces-toiminnon (kuva 9) ja Help-valikko About-toiminnon (kuva 10).



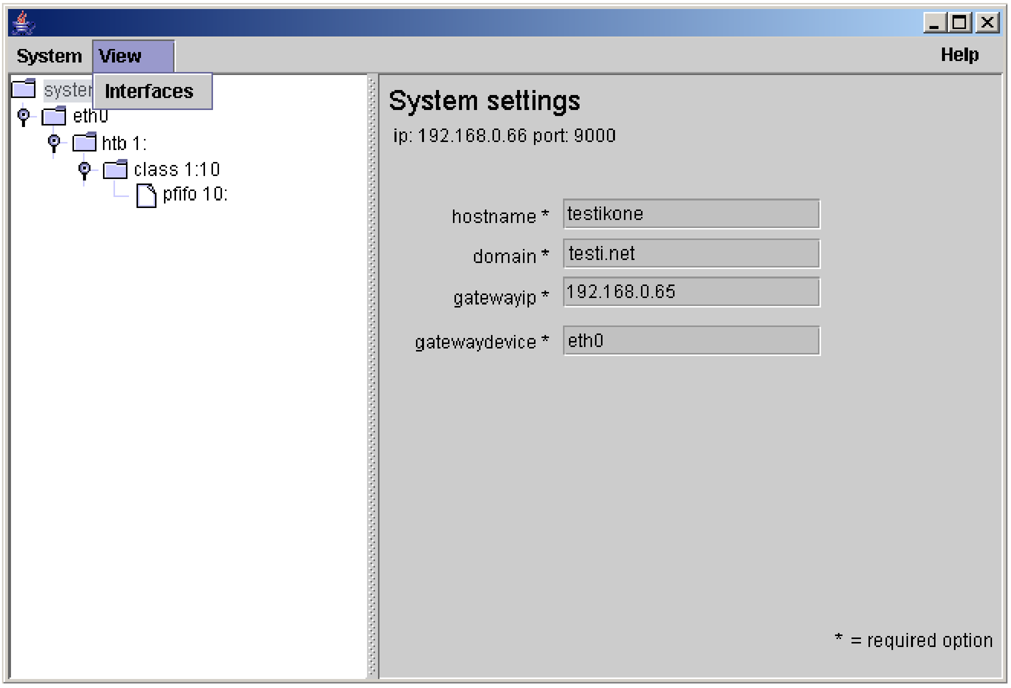
Kuva 8: Käyttöliittymä: System-valikko

Commit

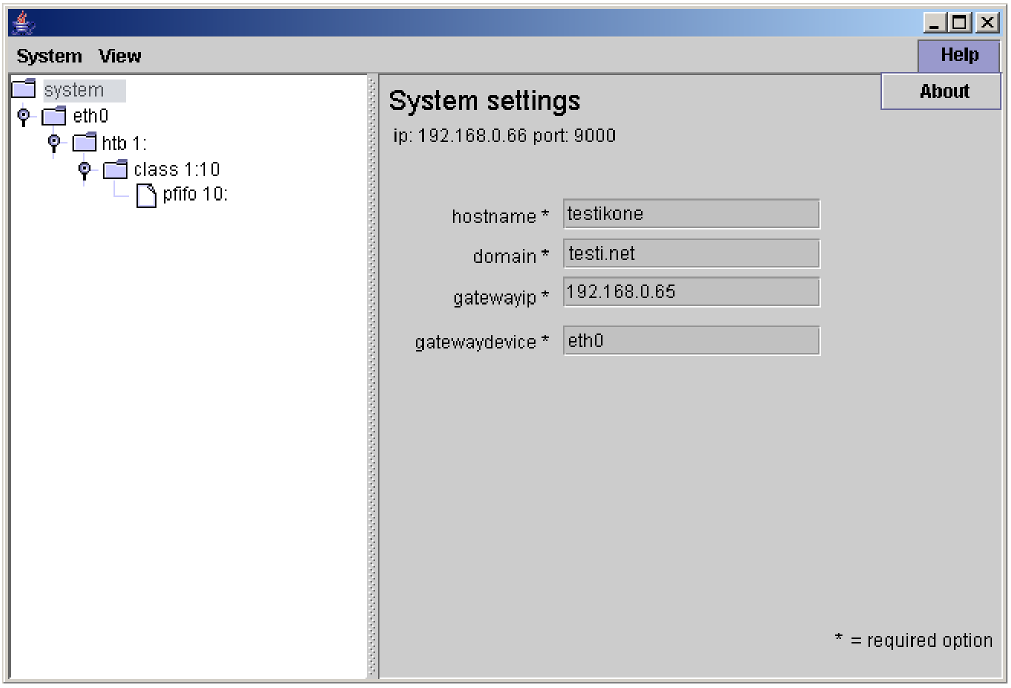
Commit-toiminnolla käyttöliittymä lähettää kaistanhallinta-asetukset kohdejärjestelmälle, joka ottaa asetukset käyttöön. Käyttöliittymä lukitsee itsensä kunnes kohdejärjestelmä lähettää kuittauksen (kuva 11). Virhetilanteissa käyttöliittymä näyttää virheilmoituksen (kuva 12). Kohdejärjestelmän vastattua käyttöliittymä palaa perusnäkymään (kuva 3). Toipuminen palaamatta jäävästä viestistä toteutetaan ajastimella.

Load from daemon

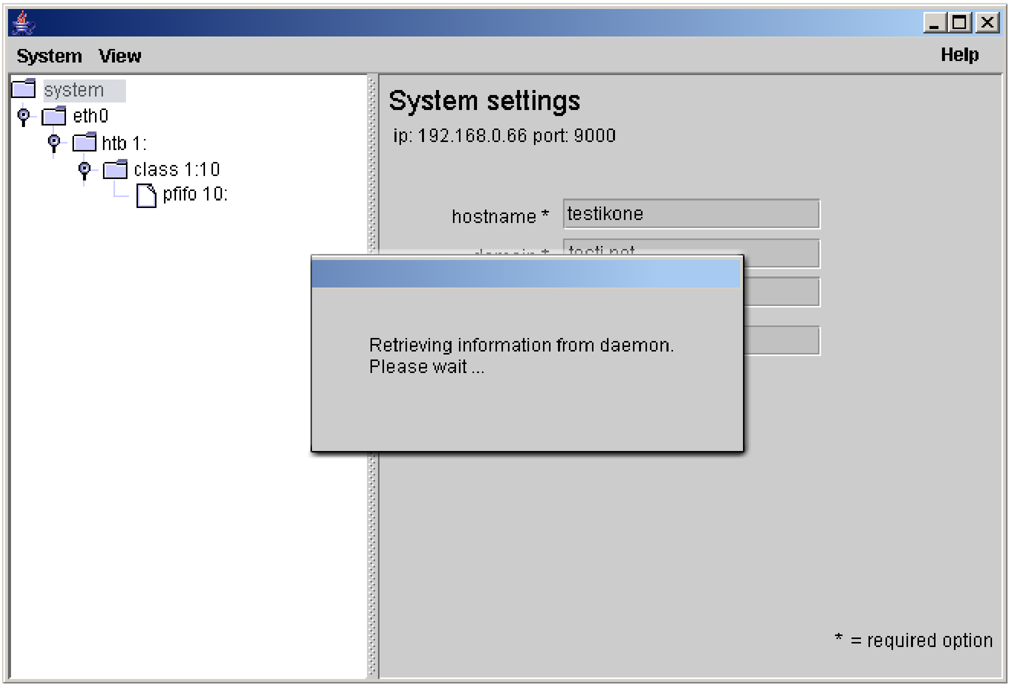
Load from daemon -toiminnolla käyttöliittymä pyytää kaistanhallinta-asetukset kohdejärjestelmältä. Käyttöliittymä lukitsee itsensä kunnes kohdejärjestelmä vastaa pyyntöön (kuva 11). Jos kaistanhallinta-asetusten haku



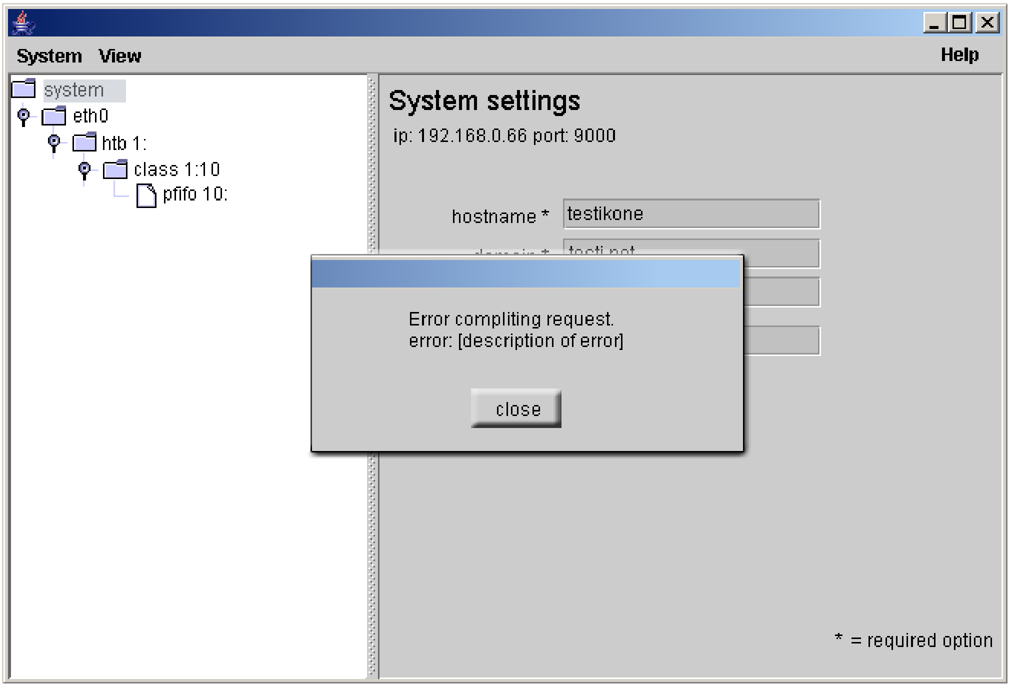
Kuva 9: Käyttöliittymä: View-valikko



Kuva 10: Käyttöliittymä: Help-valikko



Kuva 11: Käyttöliittymä: tiedonsiirron odottaminen.

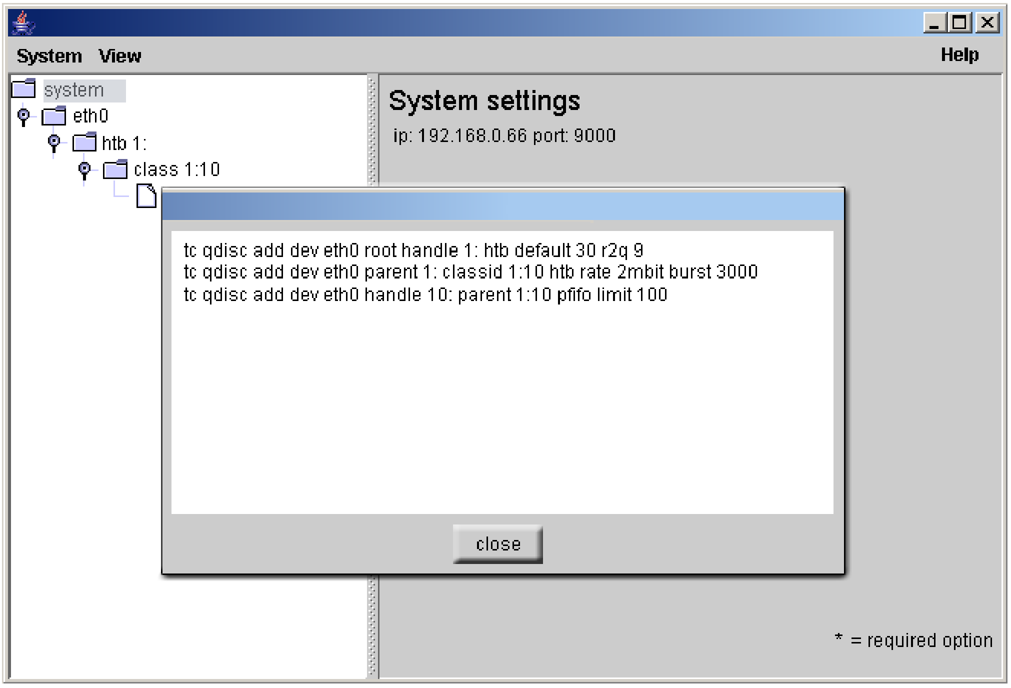


Kuva 12: Käyttöliittymä: virhetilanne

onnistuu, päivittyy puunäkymä asetusten mukaiseksi ja edelliset asetukset häviävät (kuva 3). Virhetilanteissa käyttöliittymä näyttää virheilmoituksen (kuva 12). Toipuminen palaamatta jäävästä viestistä toteutetaan ajastimella.

View as script

View as script -toiminnolla käyttöliittymä näyttää puunäkymän kohdejärjestelmän komentomuodossa. Käyttöliittymä lukitsee itsensä kunnes kohdejärjestelmä on lähettänyt pyydetyt komennot (kuva 11). Käyttöliittymä näyttää komennot omassa ikkunassa, josta ne voidaan kopioida talteen (kuva 13). Virhetilanteissa käyttöliittymä näyttää virheilmoituksen (kuva 12). Toipuminen palaamatta jäävästä viestistä toteutetaan ajastimella.



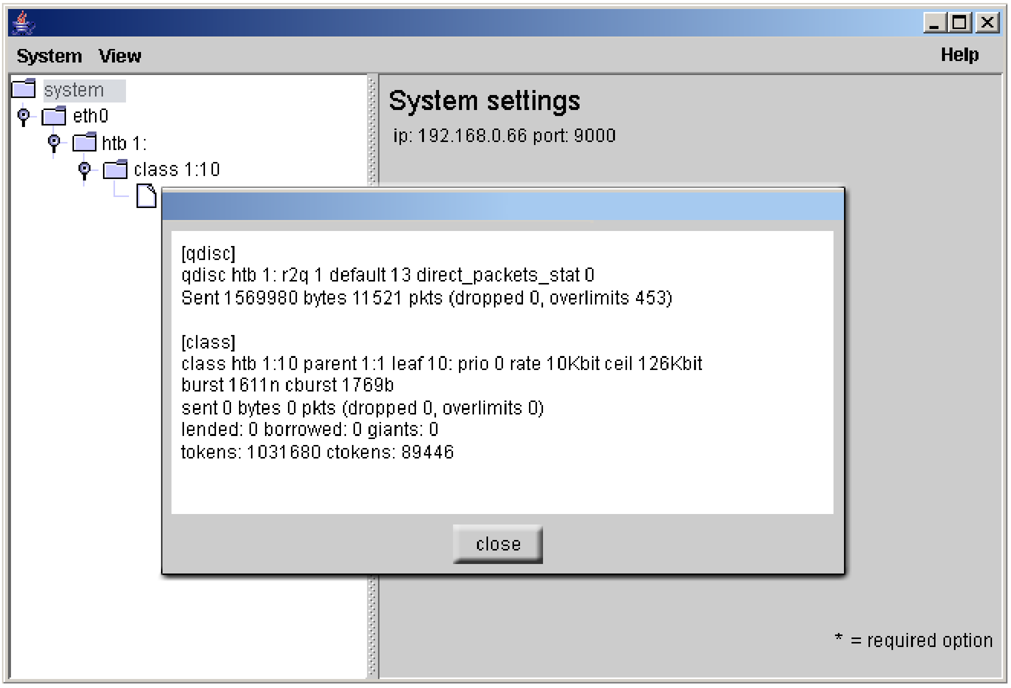
Kuva 13: Käyttöliittymä: kaistanhallinta-asetukset tc-komentomuodossa.

Show stats

Show stats-toiminnolla käyttöliittymä näyttää kohdejärjestelmän tilastotietoja. Käyttöliittymä lukitsee itsensä kunnes kohdejärjestelmä on lähettänyt pyydetyt tiedot (kuva 11). Käyttöliittymä näyttää tilastotiedot omassa ikkunassa, josta ne voidaan kopioida talteen (kuva 14). Virhetilanteissa käyttöliittymä näyttää virheilmoituksen (kuva 12). Toipuminen palaamatta jäävästä viestistä toteutetaan ajastimella.

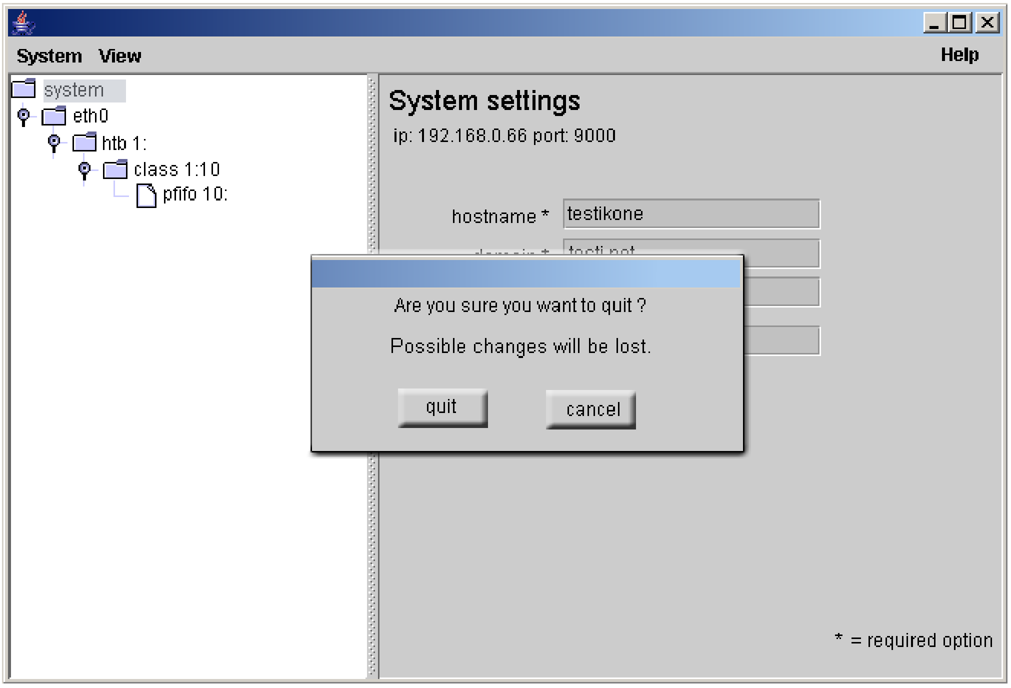
Quit

Ohjelma suljetaan Quit-toiminnolla. Mikäli käyttäjä on tehnyt muutoksia puunäkymään, käyttäjältä varmistetaan haluaako hän varmasti lopettaa (ku-



Kuva 14: Käyttöliittymä: tilastotiedot

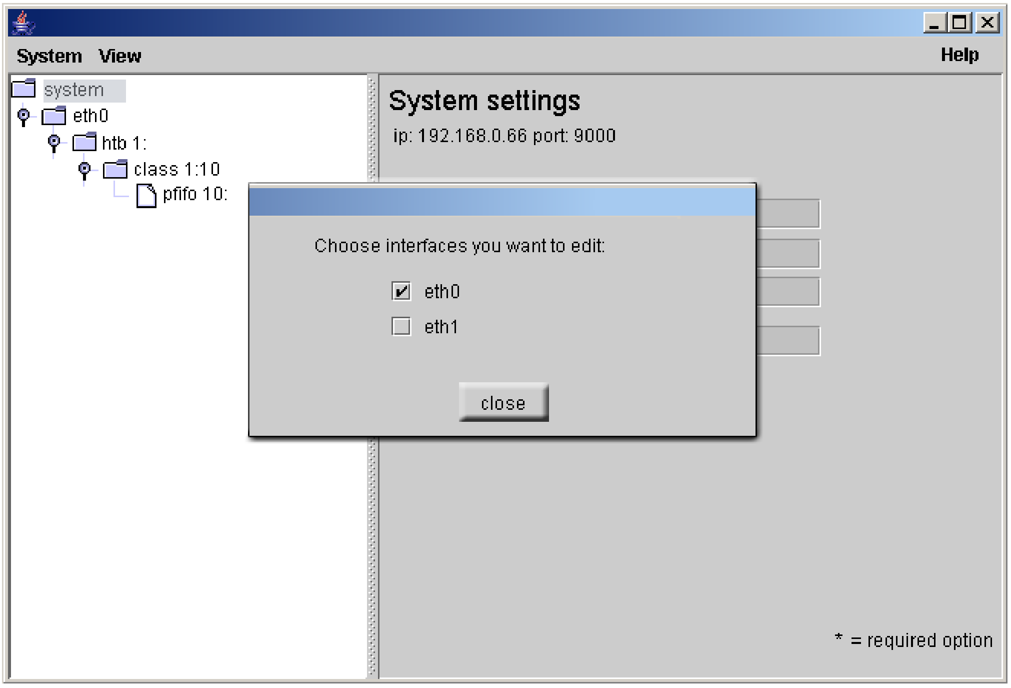
va 15). Valintaikkunan Quit-painike sulkee ohjelman tallentamatta muutoksia ja Cancel-painike palauttaa käyttöliittymän perusnäkymään (kuva 3).



Kuva 15: Käyttöliittymä: lopettaminen

Interfaces

Interfaces-toiminto avaa valintaikkunan, jossa käyttäjä voi valita minkä verkkoliittymien asetuksia hän haluaa muokata (kuva 16). Käyttäjä valitsee verkkoliittymät rastittamalla hiiren vasemmalla painikkeella verkkoliittymien ruutuja ja valitut verkkoliittymät päivittyvät puunäkymään (kuva 17).



Kuva 16: Käyttöliittymä: verkkoliittymien valinta 1.

About

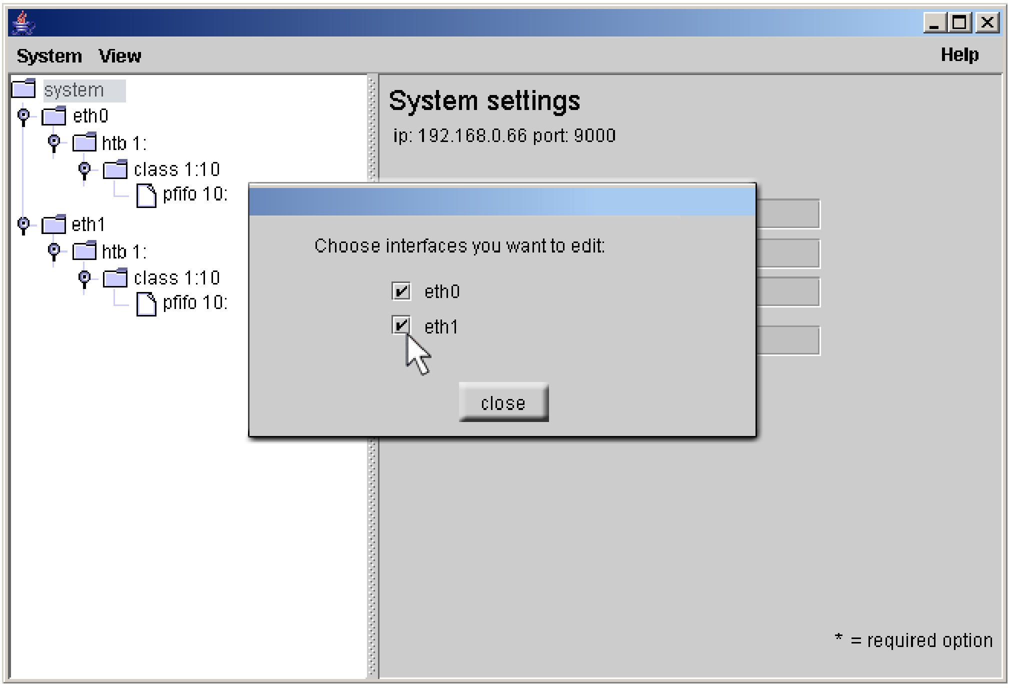
About-toiminto näyttää tietoa ohjelmasta sekä sen tekijöistä omassa ikkunassa (kuva 18).

5.3 Rakenne

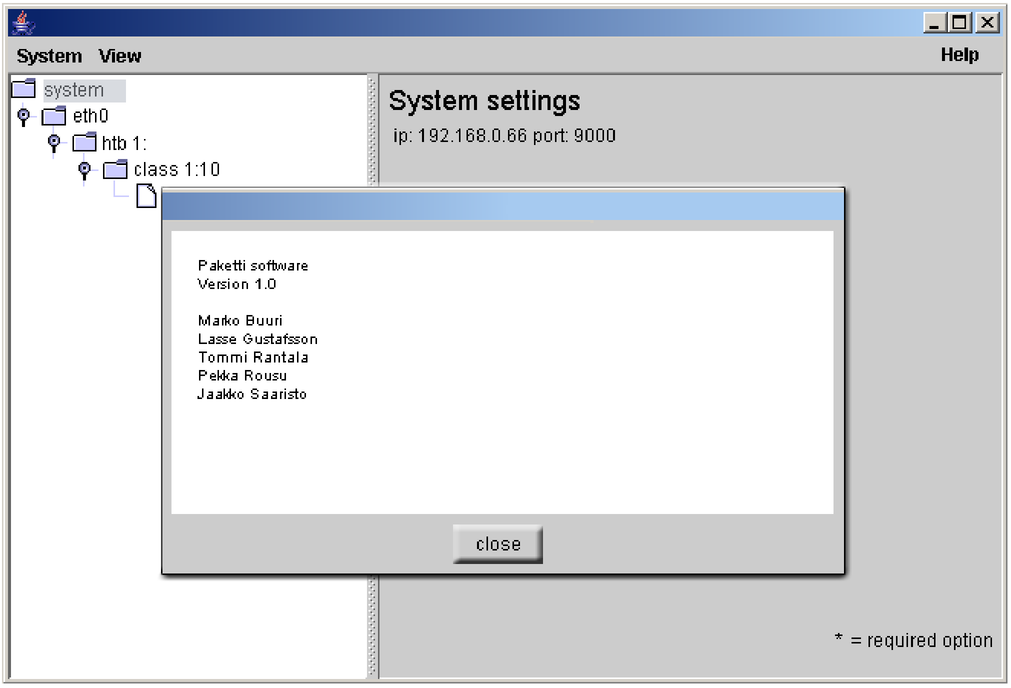
Käyttöliittymän rakenne jakautuu kolmeen osaan: view, connection ja model. PakettiGUI-luokka on ohjelman pääluokka, jonka kautta ohjelma käynnistetään. Rakenne esitetään kuvassa 19.

5.3.1 View

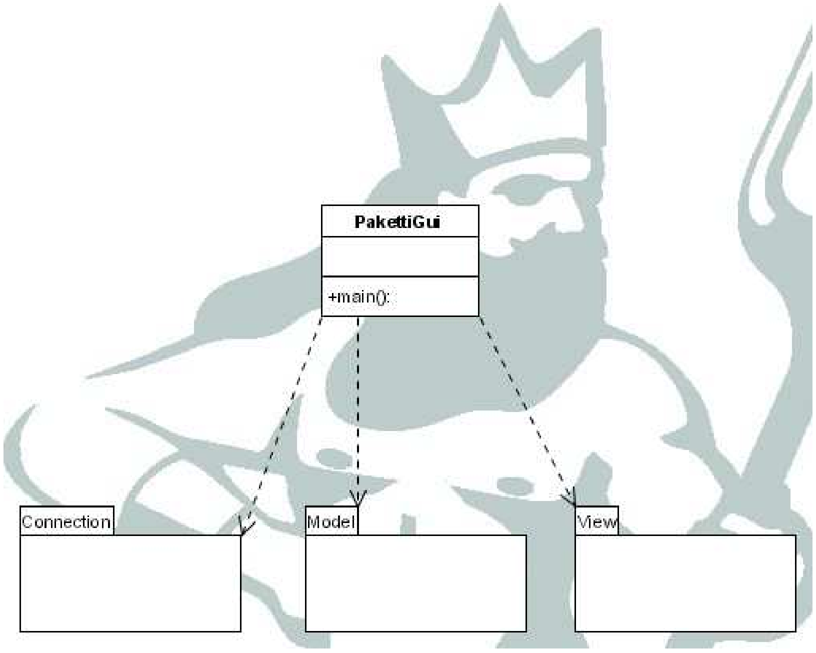
View-luokka toteuttaa graafisen käyttöliittymän. Käyttöliittymä luodaan käyttäen Javan Swing-komponentteja. View-luokka perii JFrame-luokan, joka toimii kehyksenä koko graafiselle käyttöliittymälle. View-luokka toteutettaa puu-



Kuva 17: Käyttöliittymä: verkkoliittymien valinta 2.



Kuva 18: Käyttöliittymä: tekijöiden tiedot.



Kuva 19: Käyttöliittymän rakenteen UML-kuvaus.

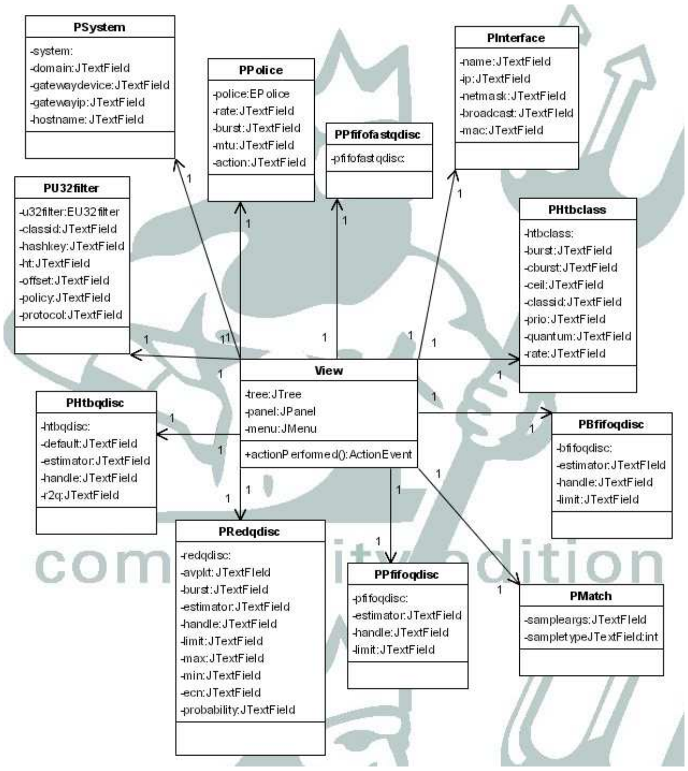
näkymän JTree-komponentilla, joka alustetaan Model-luokan määrittelemällä TreeModel-tietorakenteella. Jokaisella puunäkymän elementillä on oma asetusnäkymä, jotka toteutetaan omiin luokkiin. Asetusnäkymäluokat perivät JPanel-luokan ja esittävät asetukset JTextField-komponenteilla. Asetusnäkymäluokkia ovat PInterface, PHtbclass, PPfifofastqdisc, PPolice, PSystem, PU32filter, PHtbqdisc, PRedqdisc, PPfifoqdisc, PMatch ja PBfifoqdisc. Kuvassa 20 on esitetty View-luokan UML-kuvaus.

5.3.2 Connection

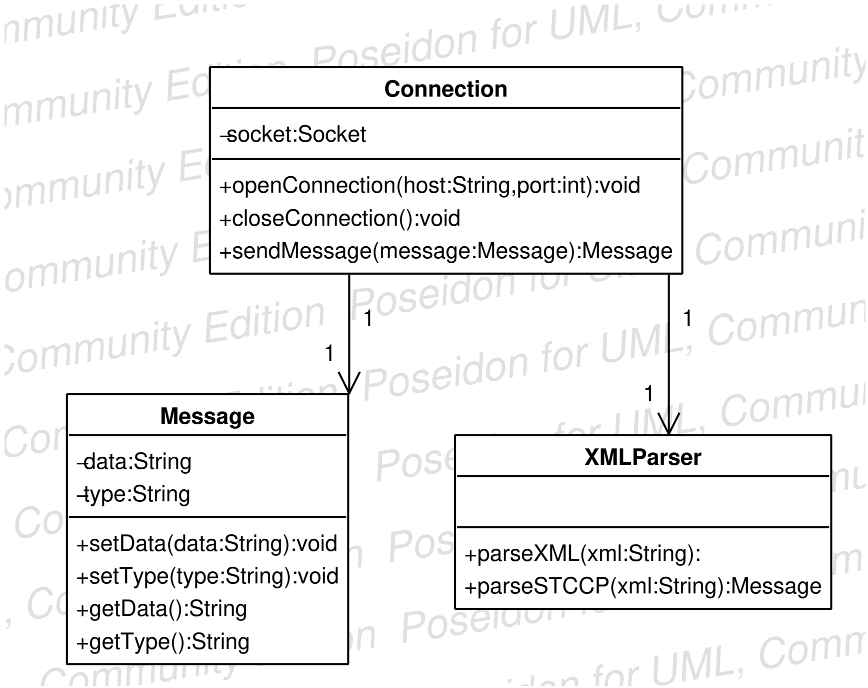
Connection-luokka toteuttaa kaiken tietoliikenteeseen liittyvän toiminnon. Connection-luokka toteuttaa TCP/IP-yhteyden luomisen ja katkaisemisen. Se toteuttaa STCCP-protokollan mukaisen viestin lähettämisen kohdejärjestelmään sekä viestin vastaanottamisen kohdejärjestelmältä. Message-luokka kuvaa STCCP-protokollan mukaista viestiä. STCCP-viestin jäsentämiseen käytetään XMLParser-luokan toteuttamia metodeja. Kuvassa 21 on esitetty Connection-luokan UML-kuvaus.

5.3.3 Model

Model-luokka toteuttaa käyttöliittymän sisäisen tietorakenteen, jolla kuvataan kaistanhallinta-asetuksia. Tietorakenne toteuttaa TreeModel-rajapinnan.

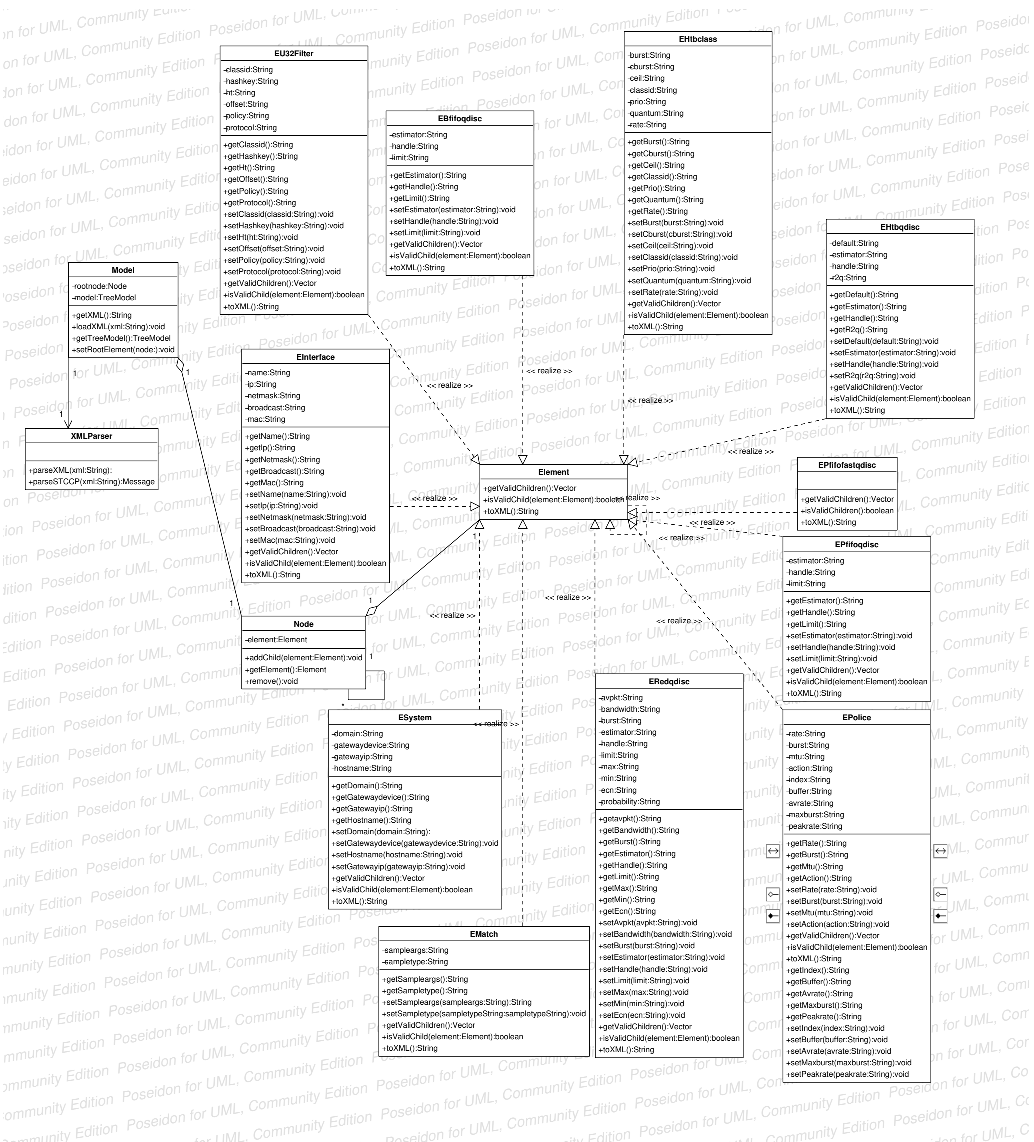


Kuva 20: View-luokan UML-kuvaus.



Kuva 21: Connection-luokan UML-kuvaus

Model-luokka toteuttaa tietorakenteen muunnokset XML-tiedostoksi sekä toisinpäin käyttäen XMLParser-luokan toteuttamia metodeja. Tietorakenne koostuu Node-luokista. Node-luokista muodostuu linkitetty rakenne, jolloin jokaisella Node-luokalla voi olla viite toisiin Node-luokkiin. Näitä viitteitä kutsutaan lapsiksi. Jokainen Node-luokka pitää sisällään ilmentymän yhdestä Element-rajapinnan toteuttavasta luokasta. Kaistanhallinta-asetukset koostuvat elementtiluokista. Jokainen elementtiluokka perii Element-rajapinnan. Elementtiluokkia ovat ESystem, EInterface, EMatch, ERedqdisc, EPfifoqdisc, EPolice, EPfifofastqdisc, EHtbqdisc, EHtbclass, EBfifoqdisc ja EU32filter. Kuvassa 22 on esitetty Model-luokan UML-kuvaus.



Kuva 22: Model-luokan UML-kuvaus

5.4 Luokat ja metodit

Seuraavassa esitetetään käyttöliittymäkomponentin luokkien konstruktorit ja metodit.

5.4.1 PakettiGUI

PakettiGUI-luokan kautta käynnistetään käyttöliittymä. Luo ilmentymät View-, Model- ja Connection-luokista.

Konstruktorit

`public PakettiGUI()`

Metodit

`public static void main(String[] args)`

Muodostaa yhteyden kohdejärjestelmään ja käynnistää graafisen käyttöliittymän.

5.4.2 Model

Model-luokka toteuttaa käyttöliittymän sisäisen tietorakenteen, joka koostuu kaistanhallinta-asetuksista.

Konstruktorit

`public Model()`

Metodit

`public String getXML()`

Muodostaa tietorakenteesta XML-tiedoston, joka kuvaa kaistanhallinta-asetuksia.

`public void loadXML(String xml)`

Muodostaa annetusta XML-tiedostosta tietorakenteen käyttäen XMLParser-luokkaa.

`public TreeModel getTreeModel()`

Palauttaa ilmentymän TreeModel:sta, jolla alustetaan View-luokan JTree-komponentti.

`public void setRootElement(Node node)`

Alustaa tietorakenteen annetulla Node-ilmentymällä.

5.4.3 Node

Sisäinen tietorakenne koostuu Node-luokista, joista koostuu puumainen rakenne. Jokainen Node-ilmentymä pitää sisällään yhden ilmentymän jostain Element-rajapinnan toteuttavasta luokasta.

Konstruktorit

`public Node(Element element)`

Metodit

`public void addChild(Element element)`

Lisää lapsen, joka pitää tiedon Element-ilmentymästä.

`public void remove()`

Poistaa itsensä sekä kaikki omat lapsensa tietorakenteesta.

`public Element getElement()`

Palauttaa ilmentymän Element:stä.

5.4.4 XMLParser

XMLParser-luokka toteuttaa XML-tiedostojen jäsentämisen.

Metodit

`public static Node parseXML(String xml)`

Muodostaa XML-tiedostosta tietorakenteen käyttäen Node-luokkaa. Palauttaa Node-ilmentymän, joka on tietorakenteen juuri.

`public static Message parseSTCCP(String xml)`

Muuntaa STCCP-protokollan mukaisen viestin Message-ilmentymäksi.

5.4.5 Connection

Connection-luokka muodostaa yhteyden kohdejärjestelmään sekä hoitaa viestien välityksen kohdejärjestelmän ja käyttöliittymän välillä.

Konstruktorit

`public Connection()`

Metodit

`public void openConnection(String host, int port)`

Muodostaa TCP/IP-yhteyden parametreina annettuun osoitteeseen.

`public void closeConnection()`

Sulkee TCP/IP-yhteyden.

`public Message sendMessage(Message message)`

Lähettää parametrina annetun viestin kohdejärjestelmän kontrollikomponentille. Lähetyksen jälkeen metodi palauttaa kontrollikomponentilta saadun vastauksen. Vastaus muunnetaan Message-luokan ilmentymäksi XMLParser-luokan avulla.

5.4.6 Message

Message-luokka kuvaa STCCP-protokollan mukaisen viestin.

Konstruktorit

`public Message(String type, String data)`

Metodit

`public String getType()`

Palauttaa viestin tyypin.

`public String getData()`

Palauttaa viestin sisällön.

5.4.7 View

View-luokka muodostaa graafisen käyttöliittymän käyttämällä Javan Swing-komponentteja. Luokan kautta hallitaan koko ohjelman toimintaa.

Konstruktorit

`public View(Model model, Connection connection)`

Metodit

`public ActionEvent actionPerformed()`

Metodia kutsutaan aina kun käyttäjä tekee toiminnon graafisessa käyttöliittymässä. Metodissa käsitellään jokainen tällainen toiminto.

5.4.8 PHtbclass

htbclass-elementin attribuuttipaneeli. Perii JPanel-luokan.

Konstruktorit

`public PHtbclass(EHtbclass htbclass)`

5.4.9 PFifofastqdisc

pfifofastqdisc-elementin attribuuttipaneeli. Perii JPanel-luokan.

Konstruktorit

`public PFifofastqdisc(EPfifofastqdisc pfifofastqdisc)`

5.4.10 PPolice

police-elementin attribuuttipaneeli. Perii JPanel-luokan.

Konstruktorit

`public PPolice(EPolice police)`

5.4.11 PSystem

system-elementin attribuuttipaneeli. Perii JPanel-luokan.

Konstruktorit

`public PSystem(ESystem system)`

5.4.12 PU32filter

u32filter-elementin attribuuttipaneeli. Perii JPanel-luokan.

Konstruktorit

`public PU32filter(EU32filter u32filter)`

5.4.13 PHtbqdisc

htbqdisc-elementin attribuuttipaneeli. Perii JPanel-luokan.

Konstruktorit

`public PHtbqdisc(EHtbqdisc htbqdisc)`

5.4.14 PRedqdisc

redqdisc-elementin attribuuttipaneeli. Perii JPanel-luokan.

Konstruktorit

`public PRedqdisc(ERedqdisc redqdisc)`

5.4.15 PPfifoqdisc

pfifoqdisc-elementin attribuuttipaneeli. Perii JPanel-luokan.

Konstruktorit

`public PPfifoqdisc(EPfifoqdisc pfifoqdisc)`

5.4.16 PMatch

match-elementin attribuuttipaneeli. Perii JPanel-luokan.

Konstruktorit

`public PMatch(EMatch match)`

5.4.17 PBfifoqdisc

bfifoqdisc-elementin attribuuttipaneeli. Perii JPanel-luokan.

Konstruktorit

`public PBfifoqdisc(EBfifoqdisc bfifoqdisc)`

5.4.18 Element

Element on rajapintaluokka, jonka jokainen elementtiä kuvaava luokka toteuttaa.

Metodit

`public Vector getValidChildren()`

Palauttaa Vector-tietorakenteessa luokkatyypit, joita saa lisätä luokan lapsiksi.

`public boolean isValidChild(Element element)`

Palauttaa arvon "true", jos parametrinä annettu luokka voidaan listätä lapseksi. Muuten palauttaa "false".

`public String toXML()`

Palauttaa luokan kuvauksen merkkijonona, joka on XML-muodossa.

5.4.19 ESystem

Luokka ESystem kuvaa system-elementtiä (luku 3.2) ja toteuttaa Element-rajapinnan.

Konstruktorit

`public ESystem(String hostname, String domain, String gatewayip, String gatewaydevice)`

Metodit

`public String getHostname()`

Palauttaa hostname-muuttujan arvon.

`public String getDomain()`

Palauttaa domain-muuttujan arvon.

`public String getGatewayip()`

Palauttaa gatewayip-muuttujan arvon.

`public String getGatewaydevice()`

Palauttaa gatewaydevice-muuttujan arvon.

`public void setHostname(String hostname)`

Asettaa hostname-muuttujalle uuden arvon.

`public void setDomain(String domain)`

Asettaa domain-muuttujalle uuden arvon.

`public void setGatewayip(String gatewayip)`

Asettaa gatewayip-muuttujalle uuden arvon.

`public void setGatewaydevice(String gatewaydevice)`

Asettaa gatewaydevice-muuttujalle uuden arvon.

5.4.20 EMatch

Luokka EMatch kuvaa match-elementtiä (luku 3.2) ja toteuttaa Element-rajapinnan.

Konstruktorit

`public EMatch(String sampletype, String sampleargs)`

Metodit

`public String getSampletype()`

Palauttaa sampletype-muuttujan arvon.

`public String getSampleargs()`

Palauttaa sampleargs-muuttujan arvon.

`public void setSampletype()`

Asettaa sampletype-muuttujalle uuden arvon.

`public void setSampleargs()`

Asettaa sampleargs-muuttujalle uuden arvon.

5.4.21 ERedqdisc

Luokka ERedqdisc kuvaa redqdisc-elementtiä (luku 3.2) ja toteuttaa Element-rajapinnan.

Konstruktorit

`public ERedqdisc(String handle, String estimator, String limit, String min, String max, String avpkt, String burst, String probability, String bandwidth, String ecn)`

Metodit

`public String getHandle()`

Palauttaa handle-muuttujan arvon.

`public String getEstimator()`

Palauttaa estimator-muuttujan arvon.

`public String getLimit()`

Palauttaa limit-muuttujan arvon.

`public String getMin()`

Palauttaa min-muuttujan arvon.

`public String getMax()`

Palauttaa max-muuttujan arvon.

`public String getAvpkt()`

Palauttaa avpkt-muuttujan arvon.

`public String getBurst()`

Palauttaa burst-muuttujan arvon.

`public String getProbability()`

Palauttaa probability-muuttujan arvon.

`public String getBandwidth()`

Palauttaa bandwidth-muuttujan arvon.

`public String getEcn()`

Palauttaa ecn-muuttujan arvon.

`public void setHandle(String handle)`

Asettaa handle-muuttujalle uuden arvon.

`public void setEstimator(String estimator)`

Asettaa estimator-muuttujalle uuden arvon.

`public void setLimit(String limit)`

Asettaa limit-muuttujalle uuden arvon.

`public void setMin(String min)`

Asettaa min-muuttujalle uuden arvon.

`public void setMax(String max)`

Asettaa max-muuttujalle uuden arvon.

`public void setAvpkt(String avpkt)`

Asettaa avpkt-muuttujalle uuden arvon.

`public void setBurst(String burst)`

Asettaa burst-muuttujalle uuden arvon.

`public void setProbability(String probability)`

Asettaa probability-muuttujalle uuden arvon.

`public void setBandwidth(String badwidth)`

Asettaa bandwidth-muuttujalle uuden arvon.

`public void setEcn(String ecn)`

Asettaa ecn-muuttujanlle uuden arvon.

5.4.22 EPfifoqdisc

Luokka EPfifoqdisc kuvaa pfifoqdisc-elementtiä (luku 3.2) ja toteuttaa Element-rajapinnan.

Konstruktorit

`public EPfifoqdisc(String handle, String estimator, String limit)`

Metodit

`public String getHandle()`

Palauttaa handle-muuttujan arvon.

`public String getEstimator()`

Palauttaa estimator-muuttujan arvon.

`public String getLimit()`

Palauttaa limit-muuttujan arvon.

`public void setHandle(String handle)`

Asettaa handle-muuttujalle uuden arvon.

`public void setEstimator(String estimator)`

Asettaa handle-muuttujalle uuden arvon.

`public void setLimit(String limit)`

Asettaa limit-muuttujalle uuden arvon.

5.4.23 EBfifoqdisc

Luokka EBfifoqdisc kuvaa bfifoqdisc-elementtiä (luku 3.2) ja toteuttaa Element-rajapinnan.

Konstruktorit

`public EBfifoqdisc(String handle, String estimator, String limit)`

Metodit

`public String getHandle()`

Palauttaa handle-muuttujan arvon.

`public String getEstimator()`

Palauttaa estimator-muuttujan arvon.

`public String getLimit()`

Palauttaa limit-muuttujan arvon.

`public void setHandle(String handle)`

Asettaa handle-muuttujalle uuden arvon.

`public void setEstimator(String estimator)`

Asettaa handle-muuttujalle uuden arvon.

`public void setLimit(String limit)`

Asettaa limit-muuttujalle uuden arvon.

5.4.24 EPolice

Luokka EPolice kuvaa police-elementtiä (luku 3.2) ja toteuttaa Element-rajapinnan.

Konstruktorit

`public EPolice(String rate, String burst, String mtu, String action)`

Metodit

`public String getRate()`

Palauttaa rate-muuttujan arvon.

`public String getBurst()`

Palauttaa burst-muuttujan arvon.

`public String getMtu()`

Palauttaa mtu-muuttujan arvon.

`public String getAction()`

Palauttaa action-muuttujan arvon.

`public void setRate(String rate)`

Asettaa rate-muuttujalle uuden arvon.

`public void setBurst(String burst)`

Asettaa burst-muuttujalle uuden arvon.

`public void setMtu(String mtu)`

Asettaa mtu-muuttujalle uuden arvon.

`public void setAction(String action)`

Asettaa action-muuttujalle uuden arvon.

5.4.25 EHtbqdisc

Luokka EHtbqdisc kuvaa htbqdisc-elementtiä (luku 3.2) ja toteuttaa Element-rajapinnan.

Konstruktorit

`public EHtbqdisc(String handle, String estimator, String default, String r2q)`

Metodit

`public String getHandle()`

Palauttaa handle-muuttujan arvon.

`public String getEstimator()`

Palauttaa estimator-muuttujan arvon.

`public String getDefault()`

Palauttaa default-muuttujan arvon.

`public String getR2q()`

Palauttaa r2q-muuttujan arvon.

`public void setHandle(String handle)`

Asettaa handle-muuttujalle uuden arvon.

`public void setEstimator(String estimator)`

Asettaa estimator-muuttujalle uuden arvon.

`public void setDefault(String default)`

Asettaa default-muuttujalle uuden arvon.

`public void setR2q(String r2q)`

Asettaa r2q-muuttujalle uuden arvon.

5.4.26 EPfifofastqdisc

Luokka EPfifofastqdisc kuvaa pfifofastqdisc-elementtiä (luku 3.2) ja toteuttaa Element-rajapinnan.

Konstruktorit

`public EPfifofastqdisc()`

5.4.27 EHtbclass

Luokka Ehtbclass kuvaa htbclass-elementtiä (luku 3.2) ja toteuttaa Element-rajapinnan.

Konstruktorit

`public EHtbclass(String classid, String rate, String burst, String ceil, String cburst, String prio, String quantum)`

Metodit

`public String getClassid()`

Palauttaa classid-muuttujan arvon.

`public String getRate()`

Palauttaa rate-muuttujan arvon.

`public String getBurst()`

Palauttaa burst-muuttujan arvon.

`public String getCeil()`

Palauttaa ceil-muuttujan arvon.

`public String getCburst()`

Palauttaa cburst-muuttujan arvon.

`public String getPrio()`

Palauttaa prio-muuttujan arvon.

`public String getQuantum()`

Palauttaa quantum-muuttujan arvon.

`public void setClassid(String classid)`

Asettaa classid-muuttujalle uuden arvon.

`public void setRate(String rate)`

Asettaa rate-muuttujalle uuden arvon.

`public void setBurst(String burst)`

Asettaa burst-muuttujalle uuden arvon.

`public void setCeil(String ceil)`

Asettaa ceil-muuttujalle uuden arvon.

`public void setCburst(String cburst)`

Asettaa cburst-muuttujalle uuden arvon.

`public void setPrio(String prio)`

Asettaa prio-muuttujalle uuden arvon.

`public void setQuantum(String quantum)`

Asettaa quantum-muuttujalle uuden arvon.

5.4.28 EInterface

Luokka EInterface kuvaa interface-elementtiä (luku 3.2) ja toteuttaa Element-rajapinnan.

Konstruktorit

`public EInterface(String name, String ip, String netmask, String broadcast, String mac)`

Metodit

`public String getName()`

Palauttaa name-muuttujan arvon.

`public String getIp()`

Palauttaa ip-muuttujan arvon.

`public String getNetmask()`

Palauttaa netmask-muuttujan arvon.

`public String getBroadcast()`

Palauttaa broadcast-muuttujan arvon.

`public String getMac()`

Palauttaa mac-muuttujan arvon.

`public void setName()`

Asettaa name-muuttujalle uuden arvon.

`public void setIp()`

Asettaa ip-muuttujalle uuden arvon.

`public void setNetmask()`

Asettaa netmask-muuttujalle uuden arvon.

`public void setBroadcast()`

Asettaa broadcast-muuttujalle uuden arvon.

`public void setMac()`

Asettaa mac-muuttujalle uuden arvon.

5.4.29 EU32filter

Kuvaa u32filter-elementtiä ja toteuttaa Element-rajapinnan.

Konstruktorit

```
public EU32filter(String protocol, String classid, String police,
String offset, String ht, String hashkey)
```

Metodit

`public String getProtocol()`

Palauttaa protocol-muuttujan arvon.

`public String getClassid()`

Palauttaa classid-muuttujan arvon.

`public String getPolice()`

Palauttaa police-muuttujan arvon.

`public String getOffset()`

Palauttaa offset-muuttujan arvon.

`public String getHt()`

Palauttaa ht-muuttujan arvon.

`public String getHashkey()`

Palauttaa hashkey-muuttujan arvon.

`public void setProtocol()`

Asettaa protocol-muuttujalle uuden arvon.

`public void setClassid()`

Asettaa classid-muuttujalle uuden arvon.

`public void setPolice()`

Asettaa police-muuttujalle uuden arvon.

`public void setOffset()`

Asettaa offset-muuttujalle uuden arvon.

`public void setHt()`

Asettaa ht-muuttujalle uuden arvon.

`public void setHashkey()`

Asettaa hashkey-muuttujalle uuden arvon.

6 Kontrollikomponentti

Kontrollikomponentin tehtävänä on toimia rajapintana käyttöliittymäkomponentin ja tc-työkalun välillä. Tietoliikenneyhteyksien hoitamiseen käytetään xinetd-palvelua, joten kontrollikomponenttiin ei tarvita sisäänrakennettuja tietoliikennetoimintoja. Xinetd-käynnistää kontrollikomponentin, kun käyttöliittymäkomponentti ottaa yhteyden kohdejärjestelmään.

6.1 Rajapinnat ja liitynnät

Kontrollikomponentin ulkoinen rajapinta on STCCP-protokolla. Kontrollikomponentissa on sisäisesti liitynnät XSLT-kirjastoon sekä tc-työkaluun. Kaikista näistä tulevat ja niihin lähtevät tiedot täytyy jäsentää.

STCCP-protokollan mukaisten viestien kaksi ensimmäistä kenttää jäsennetään itse. Length-kentän perusteella varataan muistitila XML-kehykselle. XML-kehyksen jäsennyksestä vastaa libxml-kirjasto.

XML-dokumenttien muunnos tc-komennoiksi tehdään libxml- ja libxslt-kirjastojen avulla. Tarkoitusta varten on kirjoitettu erillinen XSLT-tyylitiedosto. Libxslt jäsentää tyylitiedoston ja tuottaa halutut tc-komennot libxml:n jäsentämästä dokumentista.

Tc-työkalun syötteiden ja tulosteiden jäsentäjä kirjoitetaan itse.

6.2 Keskeiset toiminnot

Kontrollikomponentin keskeisin toiminnallisuus on muuntaa tietoa esitysmuodosta toiseen. Muunnoksia on kolmea eri tyyppiä: XML-muodon muunnos tc-komennoiksi, tc-komentojen muunnos XML-muotoon ja tc-työkalun asetusten muunnos tc-komennoiksi.

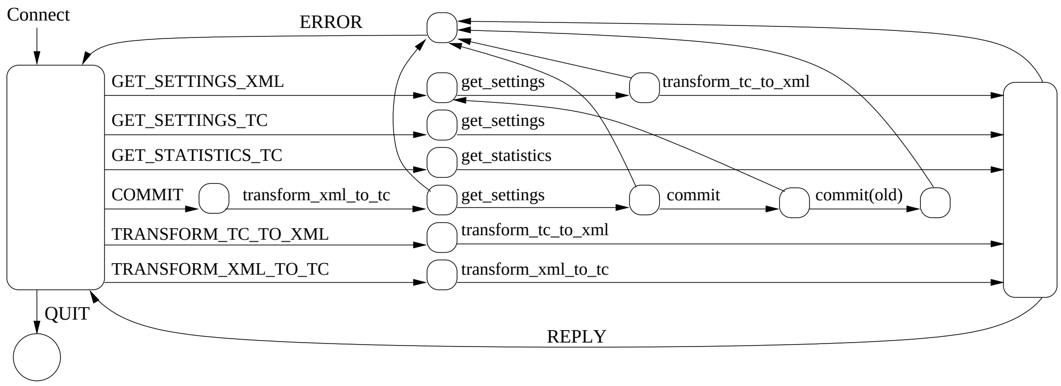
XML-dokumentti muunnetaan tc-komennoiksi kahta tarkoitusta varten:
1) kaistanhallinta-asetusten käyttöönottamiseksi ja
2) tc-komentomuotoisten asetusten esittämiseksi käyttöliittymässä.

Tc-komentoja muunnetaan XML-dokumentiksi, kun käyttöliittymä pyytää nykyisiä kaistanhallinta-asetuksia.

Tc-työkalulta pyydetyt kaistanhallinta-asetukset muunnetaan tc-komennoiksi, kun käyttöliittymä pyytää nykyisiä kaistanhallinta-asetuksia.

6.3 Tapahtumat

Kontrollikomponentti sisältää tapahtumasarjan jokaista käyttöliittymäkomponentin lähettämää viestityyppiä kohden. Tapahtumasarjojen toiminnallisuus esitetään kuvassa 23.



Kuva 23: Kontrollikomponentin tapahtumat

Seuraavassa esitetään eri STCCP-viestien käynnistämät tapahtumasarjat. Tapahtumasarjan aikana kutsuttavien funktioiden nimet on listattu sulkeissa. Kaikissa virhetilanteissa tapahtumasarjan aikana varattu muisti vapautetaan ja muodostetaan virheilmoitus, joka lähetetään send_error -funktiolla.

6.3.1 GET_SETTINGS_XML

Asetusten lataaminen tc-työkalulta (get_settings)
Asetuksien muuntaminen tc-komennoiksi (transform_settings_to_tc)
Tc-komentojen muuntaminen XML-dokumentiksi (transform_tc_to_xml)
XML-dokumentin lähettäminen käyttöliittymälle (create_xmlframe, send_reply)

6.3.2 GET_SETTINGS_TC

Asetusten lataaminen tc-työkalulta (get_settings)
Asetuksien muuntaminen tc-komennoiksi (transform_settings_to_tc)
Tc-komentojen lähettäminen käyttöliittymälle (create_xmlframe, send_reply)

6.3.3 GET_STATISTICS_XML

Pyyntöä ei ole toteutettu (send_error)

6.3.4 GET_STATISTICS_TC

Tilastotietojen lataaminen tc-työkalulta (get_statistics)
Tilastotietojen lähettäminen käyttöliittymälle (create_xmlframe, send_reply)

6.3.5 COMMIT

XML-dokumentin jäsentäminen (libxml)
XSL-dokumentin jäsentäminen (libxslt)
Muuntaminen tc-komennoiksi XSLT:n avulla (libxslt)
Komentojen siistiminen (clean_tc)
Asetusten lataaminen tc-työkalulta virheen varalta (get_settings)
Tc-komentojen suorittaminen (commit)
Mikäli tc-komentojen suorittaminen epäonnistui, palautetaan vanhat asetukset (commit)
Asetusten lataaminen tc-työkalulta (get_settings)
Asetusten muuntaminen tc-komennoiksi (transform_settings_to_tc)
Tc-komentojen muunnos XML-dokumentiksi (transform_tc_to_xml)
XML-dokumentin lähettäminen käyttöliittymälle (create_xmlframe, send_reply)

6.3.6 TRANSFORM_TC_TO_XML

Tc-komentojen muuntaminen XML-dokumentiksi (transform_tc_to_xml)
XML-dokumentin lähettäminen käyttöliittymälle (create_xmlframe, send_reply)

6.3.7 TRANSFORM_XML_TO_TC

XML-dokumentin jäsentäminen (libxml)
XSL-dokumentin jäsentäminen (libxslt)
Muuntaminen tc-komennoiksi XSLT:n avulla (transform_xml_to_tc)
Komentojen siistiminen (clean_tc)
Tc-komentojen lähettäminen käyttöliittymälle (create_xmlframe, send_reply)

6.3.8 QUIT

Kontrollikomponentin suorituksen lopetus.

6.4 Funktiot ja tiedostot

Seuraavassa esitellään kontrollikomponentin funktiot tiedostoittain ryhmiteltynä.

6.4.1 pakettid.c

Sisältää kontrollikomponentin pääohjelman.

```
int main(int argn, char *argv[])
```

Alustaa tietorakenteet, ohjaa ohjelman toimintaa ja siivoaa lopuksi tietorakenteet muistista.

6.4.2 stccp.c

Sisältää tietoliikenteeseen liittyvät funktiot.

```
xmlChar *read_request()
```

Lukee stdin:stä tietoa rivin kerrallaan, ja tarkistaa oliko syöte STCCP-protokollan mukainen. Jos ei ollut, syöte hylätään, ja funktio jatkaa lukemista. Mikäli viesti oli STCCP-protokollan määrityksen mukainen, funktio palauttaa osoittimen viestin XML-kehykseen.

```
int get_msgtype(xmlChar *xmlframe)
```

Palauttaa XML-kehyksen sisältämän pyynnön tyypin kokonaislukuna. Vaihtoehdot ovat määritelty stccp.h-tiedostossa. Virhetilanteissa funktio palauttaa arvon -1.

```
xmlChar *get_msgdata(xmlChar *xmlframe)
```

Palauttaa osoittimen XML-kehyksen data-osuuden alkuun.

```
xmlChar *create_xmlframe(int type, xmlChar *xmldata)
```

Muodostaa STCCP:n mukaisen XML-kehyksen.

```
void send_reply(xmlChar *xmlframe)
```

Lähettää XML-kehyksen STCCP-protokollan mukaisena viestinä stdout:iin.

```
void send_error(xmlChar *errormsg)
```

Lähettää virheilmoituksen STCCP-protokollan mukaisena viestinä stdout:iin.

6.4.3 tc_control.c

Sisältää tc-työkalun kanssa kommunikoivat funktiot. Tc suoritetaan popen-funktiolla, joka palauttaa FILE-osoittimen tc:n tulosteisiin. FILE suljetaan pclose-funktiolla.

```
xmlChar *commit(xmlChar *tccommands)
```

Suorittaa parametrina annetut tc-komennot. Mikäli komentojen suoritus onnistui virheittä, funktio palauttaa NULL-arvon. Virhetilanteessa funktio palauttaa tc-työkalun tulostamat ilmoitukset.

```
xmlChar *get_settings()
```

Pyytää kaistanhallinta-asetukset tc-työkalulta ja järjestää asetukset niin, ettei jonomalleja, luokkia tai suodattimia määritellä ennen niiden parent-elementtiä.

Funktio tallentaa tc-työkalun tulosteen dynaamiseen taulukkoon. Taulukon koko kaksinkertaistuu, kun tila loppuu kesken. Funktio palauttaa osoittimen taulukon alkuun.

Aluksi järjestelmältä kysytään olemassaolevat verkkoliitynnät ip-työkalulla:

```
ip link
```

Tulostetuista verkkoliitynnöistä valitaan muotoa eth([0-9]|([1-9][0-9]+)) olevat nimet. Loopback-liityntää lo ei valita. Mikäli järjestelmästä ei löydy yhtään tuettua liityntää, funktio palauttaa NULL-arvon.

Tämän jälkeen jokaista valittua verkkoliityntää kohden suoritetaan seuraavat komennot, joilla pyydetään tc-työkalulta liitynnän kaistanhallita-asetukset:

```
tc -d qdisc show dev eth0
tc -d class show dev eth0
tc -d filter show dev eth0
```

Mikäli ensimmäinen tc-komento ei tulosta mitään, jonomalliksi merkitään pfifo_fast, eikä muita komentoja suoriteta samalle verkkoliitynnälle. Luokkia ei ole määritelty, jos tc ei tulosta toisella komennolla mitään. Suodattamia ei ole määritelty, jos tc ei tulosta kolmannella komennolla mitään.

Tämän jälkeen funktio järjestää tc-työkalun tulosteet aiemmin määriteltyyn järjestykseen, ja palauttaa osoittimen merkkijonon alkuun.

```
xmlChar *get_statistics()
```

Hakee tilastotiedot tc-työkalulta ja palauttaa osoittimen tilastotietojen alkuun. Funktio tallentaa tilastotiedot dynaamiseen taulukkoon, jonka toiminta on vastaava kuin get_settings-funktiossa.

Aluksi järjestelmältä kysytään olemassaolevat verkkoliitynnät ip-työkalulla:

```
ip link
```

Tulostetuista verkkoliitynnöistä valitaan muotoa eth([0-9]|([1-9][0-9]+)) olevat nimet. Loopback-liityntä lo jätetään valitsematta. Mikäli järjestelmästä ei löydy yhtään tuettua liityntää, merkkijono jää tyhjäksi, ja funktio palauttaa osoittimen sen alkuun.

Tämän jälkeen jokaista valittua verkkoliityntää kohden suoritetaan seuraavat komennot, joilla pyydetään tc-työkalulta liitynnän kaistanhallinnan tilastotiedot:

```
tc -s -d qdisc show dev eth0
tc -s -d class show dev eth0
```

Mikäli tc-työkalu ei tulosta mitään merkkijono jää tyhjäksi. Lopuksi funktio palauttaa osoittimen merkkijonon alkuun.

6.4.4 transform.c

Sisältää kontrollikomponentin sisäiset muunnosfunktiot.

```
xmlChar *transform_xml_to_tc(xmlDocPtr xmldata,
                             xsltStylesheetPtr xmlstyle)
```

Muuntaa XML-dokumentin tc-komennoiksi käyttäen XSLT-tyylitiedostoa.

```
xmlChar *clean_tc(xmlChar *tccommands)
```

Siistii libxslt:n tuottamat tc-komennot. Komennoista poistetaan muun muassa turhat rivinvaihdot.

```
xmlChar *transform_settings_to_tc(xmlChar *tcsettings)
```

Muuntaa kaistanhallinta-asetukset tc-komennoiksi. Funktio tallentaa valmiit tc-komennot get_settingsissä esiteltyyn dynaamiseen taulukkoon ja palauttaa osoittimen taulukon alkuun.

```
xmlDocPtr transform_tc_to_xml(xmlChar *tccommands)
```

Muuntaa tc-komennot XML-dokumentiksi rekursioperiaatteella. Funktio alustaa XML-rakenteen, lisää siihen root-qdiscin ja kutsuu tc_to_xml -funktiota root-qdiscin id:llä.

```
void tc_to_xml(xmlChar *tccommands, xmlDocPtr xmldoc,
               xmlChar *parentid);
```

transform_tc_to_xml:n rekursioperiaatteella toimiva apufunktio.

1. Etsi komento, jonka id==parentid. Jos sellaista ei löydy, poistu funktiosta.
2. Lisää komento ja tee rekursiokutsu elementin id:llä.
3. Jatka kohtaan 1.

Mikäli tccommands-taulukko on järjestetty get_settings-funktiossa kuvatulla tavalla, voidaan algoritmin aikavaatimusta keventää siirtämällä rekursiokutsussa tccommands-taulukon osoitinta osoittamaan viimeiseksi lisätyn komennon loppuun.

6.4.5 header-tiedostot

Funktioiden prototyypit sijoitetaan header-tiedostoihin. Header-tiedostoihin sijoitetaan myös kontrollikomponentin osien yhteisiä määrityksiä.

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*

Kuvat

Taulukot

Viitteet

[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

[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

[PAK03a] Paketti-projektiryhmä, Määrittelydokumentti, 2003. http://www.cs.helsinki.fi/group/paketti/

[RFB01] Ramakrishnan, K., Floyd, S. ja Black, D., RFC 3168 - The Addition of Explicit Congestion Notification (ECN) to IP, 2001. http://www.ietf.org/rfc/rfc3168.txt,24.10.2003