Seramon projekti

Kimmo Haukimäki & Jani Lirkki

Agora Center
Jyväskylän yliopisto, 40351 Jyväskylä
Suomi

Contents

1 Johdanto

Seramon (Adaptive service aggregation in IP mobile networks) eli mukautuva palvelujen koostaminen mobiilissa IP verkossa. Projektin avainsana on mukautuvuus, joka tarkoittaa että verkonresurssien on mukauduttava tietoliikenteen tarpeisiin tai tietoliikenteen vaatimusten on mukauduttava verkonresurssien vaatimuksiin.

Ensimmäisessä vaihtoehdossa tekniikka kontroloi verkon resursseja sen mukaan mitä tarvitaan. QoS/CoS palveluluokkien resurssien (kaistanleveys) avulla tekniikka jakaa resursseja kullekin tietovirralle ja pyrkii näin pitämään palveluluokan mukaiset tarpeet saavutettavissa. Tällöin verkon resurssit pyritään pitämään tarpeiden mukaisina, kasvatetaan tai vähennetään kaistanleveyttä liikenteen määrän mukaan.

Jälkimmäinen kohta on edellisen vastakohta. Siinä palveluluokkien resurssien on mukauduttava siihen mitä verkon resurssit määräävät. Tämä tarkoittaa tietyn prosentuaalisen osuuden antamista verkon resursseista kullekin palveluluokalle, jolloin luokan on sopeuduttava näihin vaatimuksiin.

Projektin tarkoituksena ei ole luoda uutta siirtotekniikkaa, vaan luoda olemassa olevan tekniikan pohjalta uusi muokattu versio joka toteuttaa halutut kriteerit (mukautuva palvelujen koostaminen).

2 Systeemin kuvaus

Systeemi pohjautuu MPLS tekniikkaan, joka on luotu nopeuttamaan reitittimissä tapahtuvaa pakettien edelleenlähetystä. MPLS tekniikka on rakennettu IPv6 tekniikan päälle. MPLS tekniikasta enemmän kappaleessa 2.1.

Systeemin rakenteena on ns. kontrollisilmukka rakenne, jonka avulla hallitaan systeemin parametrejä (kaistaleveys, palveluluokkien resurssit jne). Kontrollisilmukka vertailee solmun ja verkon nykyisiä tiloja ja tekee niiden perusteella muutoksia systeemiin. Kontrollisilmukan lohkot on kuvattu kappaleessa 2.2.

2.1 MPLS tekniikka

MPLS eli MultiProtocol Label Switching on IETF:n protokolla, joka sijoittuu siirtoyhteys- ja verkkokerroksen välimaastoon [1]. MPLS tekniikan perusideana on vähentää verkkokerrosreitityksen tarvetta korvaamalla se siirtoyhteystason kytkennällä. Reititystietojen ylläpito hoidetaan MPLS tekniikassa perinteisillä verkkokerroksen reititysprotokollilla (OSPF tai BGP). Verkkokerroksenprotokollien ylläpitämien tietojen perusteella pakettiin liitetään lyhyt etiketti, jonka avulla paketti pystytään nopeasti välittämään oikealle reitille. Näin jokaisen reitittimen ei tarvitse tutkia jokaisen paketin verkkokerroksen otsikkoa, vaan ne voivat tehdä useimmiten edelleenlähetyksen suoraan siirtoyhteyskerroksen otsikkotietojen

perusteella.

Etiketti on vain lyhyt merkintätapa joukolle käyttäjien tietovirtoja, jotka kuuluvat samaan edelleenlähetys ekvivalenssiluokkaan (FEC = Forwarding Equivalent Class). Samaan ekvivalenssiluokkaan kuuluvat pakettien reititetään solmusta samalla tavalla eteenpäin.

2.2 Kontrollisilmukka

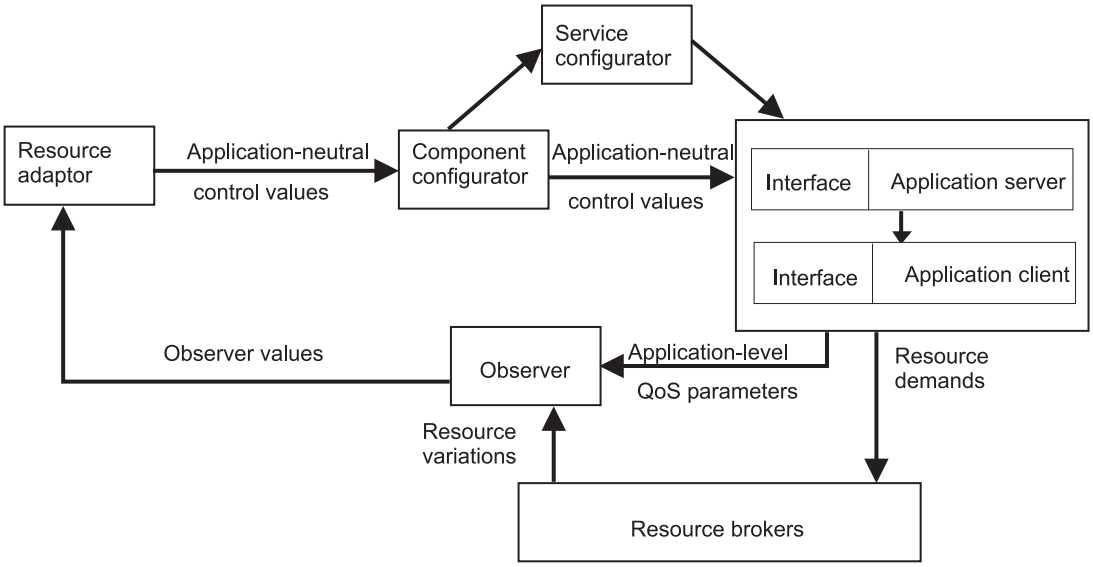

Figure 1: QoS mukautuvuus kontrollisilmukassa.

2.2.1 Data plane

Data plane on kuvassa 1 lohko, jossa on client-server rajapinta. Data plane on systeemin välittäjä, joka ottaa vastaan paketteja ja edelleenlähettää ne. Data planessa sijaitsee kaikki informaatio reitityksistä. Data plane edelleenlähettää paketteja kaistanleveydellä, joka sille sallitaan. Tiedot verkon tilasta, käytettävästä kaistanleveydestä ja pakettien lähetyskoosta se saa Service configuratorilta ja Component configuratorilta.

Data plane välittää Resource brokerille tietoa siitä kuinka paljon kaistanleveyttä tarvitaan, sekä Observerille tietoa QoS parametreista.

2.2.2 Resource brokers

Resource broker saa Data planelta tietoa kaistanleveyden vaatimuksista. Lohkossa kaistanleveys pilkotaan tarpeiden mukaisesti osiin (palveluluokkien mukaan, tai ekvivalenssi luokka kohtaisesti). Resurssien tiedot välitetään Observerille.

2.2.3 Observer

Observer saa vaatimuksien mukaiset tiedot Resource brokerilta ja QoS parametritietoja (läpimenoviive, lähetysnopeus, viiveiden muutos jne) Data planelta. Observer vertailee näitä tietoja keskenään ja välittää vertailutulokset Resource adaptorille.

2.2.4 Resource adaptor

Resource adaptor on kontrollisilmukan aivot. Se tekee päätöksen Observerilta tulevien tietojen pohjalta. Resource adaptor päättää seuraavan tilan systeemille, eli mikä on kaistanleveys, lähetettävän paketin maksimikoko, palveluluokkien resurssit jne. Tiedot välitetään Component configuratorille.

2.2.5 Component configurator

Component configuratorin tehtävä on muuttaa systeemin tilaa, joka koskee kokonais resursseja (solmun kokonais kaistanleveys). Muutokseen tarvittavat tiedot välittyvät Resource adaptorilta, ja tiedot muutoksesta välitetään Data planelle.

2.2.6 Service configurator

Service configuratorin tehtävä on muuttaa systeemin tilaa, joka koskee vuo tai luokka kohtaisia resursseja. Muutokset välitetään Data planelle.

2.3 Edelleenlähetys

Systeemi saa tietoa kokoajan verkon tilasta ja pyrkii mukautumaan parhaalla mahdollisella tavalla sen tarpeisiin. Systeemin tila on siis riippuvainen verkon tilasta, ja tilan muutos tapahtuu sen perusteella.

Systeemin tilaa vaihdetaan Markovin ketju mallin mukaan. Kuvassa 2 on esitetty edelleenlähetysnopeus verrattuna pakettien kokoon. Pisteet edustavat arvovälejä. Markovin ketju mallin mukaisesti systeemin tilaa pystytään vaihtamaan ainoastaan vaaka- tai pystysuunnassa.

Systeemillä on kaksi erityyppistä käyräjoukkoa, jotka ovat määritelty eri palveluluokille. Varma (Assured) edelleenlähetys (kuva 3) ja kiirehditty (expedited) edelleenlähetys (kuva 4). Ensimmäinen koskee palveluluokkia, jotka ovat prioriteettitasoltaan alhaisempia (2-5). Jälkimmäinen on tarkoitettu datalle, jonka viiveet täytyy olla mahdollisimman pienet ja prioriteetti on 1.

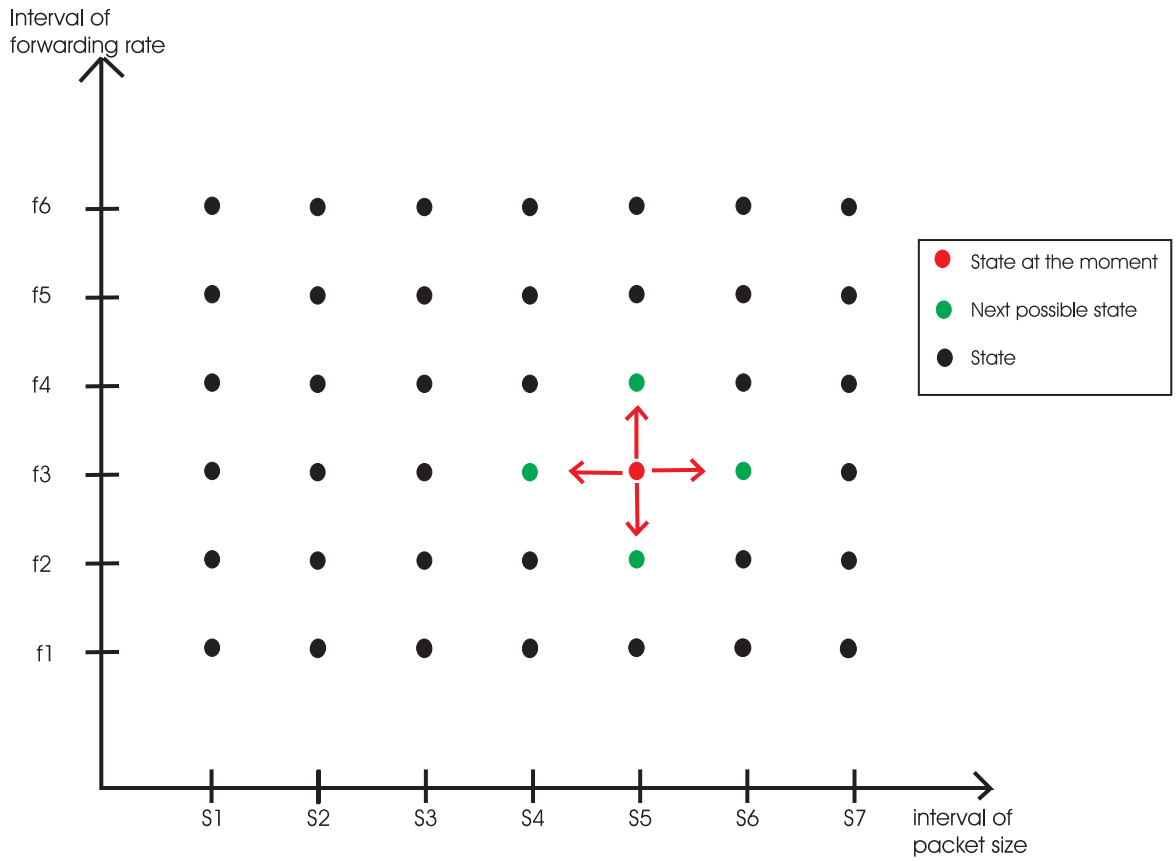

Figure 2: Markovin ketju malli edelleenlähetyksessä.

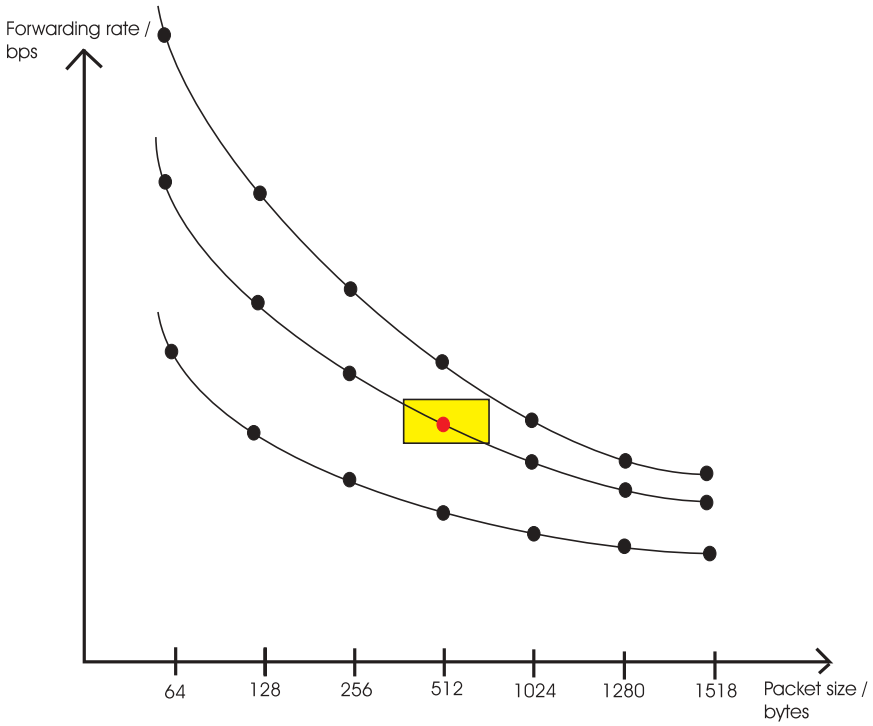

Figure 3: Varman edelleenlähetyksen nopeudet verrattuna pakettien kokoihin.

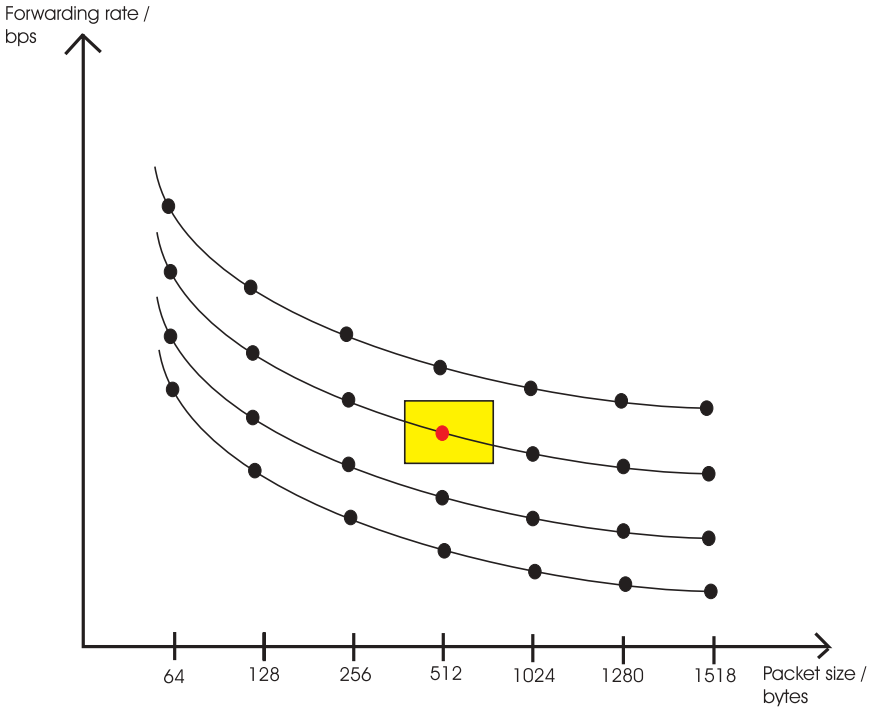

Figure 4: Kiirehdityn edelleenlähetyksen nopeudet verrattuna pakettien kokohin.

3 Toteutetut

- Multreeldp-protokollan toteutukseen on tutustuttu ja testattu. Suurin osa toteutuksesta on liitetty osaksi meidän ohjelmaamme, ja se toteuttaa multicast laajennoksen CR-LDP:lle. Nortel Networks on julkaissut vapaan LDP-protokollan, CR-LDP, ja se toteuttaa signaloinnin MPLS verkossa. Tämä vaatii MPLS-linux-kernelin, joka on yksi syy valintaamme työskennellä linux ympäristössä.

- Ensimmäinen toteutus MPLS solmun toiminnasta on valmis. IPv6 otsikon diffserv kentän muuttaminen MPLS otsikon 'experimental'-kenttään on toteutettu. **Tämä sallii liikenteen käsittelyn prioriteetin mukaan MPLS-reititimessä**.

- TSA-scheduling menetelmä, jonka Juhani Takkinen on kehittänyt, on toteutettu c++:lla ja liitetty osaksi ohjelmaamme.

- Olavi Paanasen toteuttama puskurinhallinta on liitetty ohjelmaamme.

- Mika Partanen on toteuttanut graafista esitystä ohjelmaamme varten. Graafisella versiolla pystytään esittämään joitakin haluamiamme tuloksia windowsissa.

- Ohjelma on jaettu kontrollointiviestien käsittely- ja datan käsittely säikeisiin, jotka ovat pääsäikeet. Näiden pääsäikeiden alla pyörii useampi yksittäinen säie, jotka ovat jo toiminnassa.

- Edelleenlähetyksessä on toteutettu pakettien lähetyksen kontrollointi, joka tarkoittaa kaistanleveyden kontrollointia. Kaistanleveys on verrannollinen pakettien kokoon, joka on esitetty kuvassa 3. Näistä käyristä yhdellä pystytään liikkumaan.

4 Tulevaisuuden ideat

- Välipuskurit pitäisi toteuttaa toisella tavalla, jotta voisimme käsitellä paketteja pienemmissä palasissa. Ongelmana on että tällä hetkellä pienin yksikkö on paketti (koot vaihtelevat), ja koska scheduling menetelmä jakaa resursseja yksiköiden mukaan, niin pienemmät paketit kärsivät tästä.

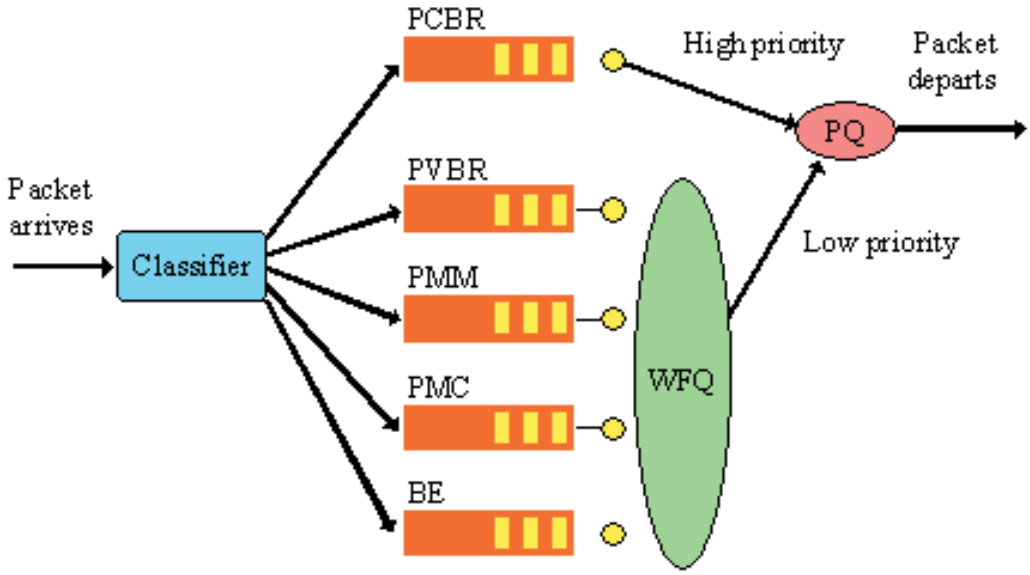

Figure 5: Luokittelu ja luokkien käsittely.

- Kuten kuvassa 5 on esitetty, korkeimman prioriteetin luokan liikenne on muutettava kulkemaan suoraan 'edelleenlähetysmoottorille'.
- Tunnelointi on toteutettu multreeldp protokollassa, mutta se tarvitsee tiedot solmuista etukäteen. Polun laskeminen ja todentaminen on toteuttamatta. Tämä voidaan tehdä E-MIRA systeemillä.
- Kuvasta 1 on dataplanen ensimmäinen toteutus valmis, mutta varsinainen QoS mekanismi on toteuttamatta. Mekanismin toteutukseen, eli kuvan 1 muihin lohkoihin ollaan juuri pääsemässä käsiksi.
- Edelleenlähetyksen osuudesta on toteuttamatta mekanismi, jolla pystytään siirtymään arvovälistä toiseen (Markovin ketju -mallin mukaisesti).

References

[1] P. Kuusi, MPLS-standardit, 1998

[2] Su-Wei Tan, Sze-Wei Lee, Benoit Vaillaint, "Non-greedy minimum interface routing algorithm for bandwitdh-quaranteed flows"; Center for High Speed Broadband Network. Multimedia University, Cyberjaya, Malesia, July 2001